HealthQ – AI Powered Healthcare Platform for Appointments & Disease Detection

By

Waresur Rahman

212-15-4235

Mahmuda Akter Mumu

213-15-4398

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Sheak Rashed Haider Noori

Professor & Head

Department of Computer Science & Engineering

Daffodil International University

Co-Supervised By

Mr. Hasanuzzaman Dipu

Assistant Professor

Department of Computer Science & Engineering

Daffodil International University

September 16, 2025

APPROVAL

This Project/internship titled **"HealthQ – AI Powered Healthcare Platform for Appointments & Disease Detection"** submitted by **"Waresur Rahman"**, ID No: 212-15-4235 and **"Mahmuda Akter Mumu"**, ID No: 213-15-4398 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 16-09-2025

BOARD OF EXAMINERS

16.09.25

Dr. Bimal Chandra Das (BCD)
Professor and Dean (in-charge) — **Chairman**
Department of Computer Science & Engineering
Faculty of Science & Information Technology
Daffodil International University

Most. Hasna Hena (HH)
Assistant Professor — **Internal Examiner**
Department of Computer Science & Engineering
Faculty of Science & Information Technology
Daffodil International University

Mr. Mayen Uddin Mojumdar (MUM)
Assistant Professor — **Internal Examiner**
Department of Computer Science & Engineering
Faculty of Science & Information Technology
Daffodil International University

Nazibur Rahman (NZR)

Technical Lead – Database Administrator — **External Examiner**
Telenor - Grameen Phone Account

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Dr. Sheak Rashed Haider Noori, Professor & Head, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised By:

15.09.25

Dr. Sheak Rashed Haider Noori

Professor & Head
Department of CSE
Daffodil International University

Co-Supervised By:

15.09.25

Mr. Hasanuzzaman Dipu

Assistant Professor
Department of CSE
Daffodil International University

Submitted By:

15.09.25

Waresur Rahman

212-15-4235
Department of CSE
Daffodil International University

Mumu 15.09.25

Mahmuda Akter Mumu

213-15-4398
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing, made us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Dr. Sheak Rashed Haider Noori,** Professor & Head, Department of CSE, Daffodil International University, Dhaka. Deep knowledge & keen interest of our supervisor in the field of "Web Development" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Sheak Rashed Haider Noori, Head, Department of CSE, for his kind help to finish our project and also to the other faculty members and the staffs of CSE department of Daffodil International University.

We would like to thank our entire course mates in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Healthcare systems spend most of the time struggling with inappropriate appointment scheduling, extensive queues, constrained physician-patient communication process, and insufficient diagnostic equipment at early stages. **HealthQ-AI Powered Healthcare Platform for Appointments & Disease Detection** is a health technology intended to solve these problems through online scheduling, real time queue prediction, AI based early detection and report management in a single web app. With this system patients are guaranteed to make appointments with the doctors of their prior choice and an artificial intelligence powered prediction model is used to estimate the length of the queue so that there is less unnecessary congestion in the hospitals and clinics. Patients can use a deep learning model to upload images of the lesion on their skin to get an initial melanoma screening that will help diagnose melanoma early and treat it promptly. Moreover, there is safe management of the reports, where one can share the reports to the doctors, and the patients can view or download at any time. Other features include pills buying and optional real time doctor patient chat module that increases user experience and engagement. HealthQ is a smart healthcare product that incorporates smart queue, AI early recognition capabilities and efficient communication to become more scalable, accessible, and user-friendly in their healthcare solution. This type of system would reduce patient waiting time, improve reliability in the diagnosis data and effectiveness in the delivery of healthcare services, and therefore availability of medical services to meet the demands of the patients and healthcare providers.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the world healthcare is a massive industry. Many developing regions, including Bangladesh healthcare system suffers from inefficiencies such as long patient queues, unoptimized appointment scheduling and limited access to early detection of diseases causing untimely death. The digital health platforms in Bangladesh does provide some facilities such as, telecommunication and report access but lacks many intelligent features mentioned above. HealthQ aims to bridge the gap. It is a modular, web-based healthcare platform designed to streamline patient services through the power of machine learning and modern web technologies. It addresses inefficiencies in appointment management and early diseases detection.

1.1 Introduction

Healthcare is an extremely important sector both for the betterment of mankind and how much value it holds. The global digital healthcare market value has risen exponentially in recent times. But in Bangladesh it is still quite far behind. Though there are many healthcare platforms in Bangladesh, many don't offer current day and age modern technology. HealthQ aims to design a web-based application of full extent that revolutionizes healthcare service delivery through understanding the inefficiencies of current healthcare platform in Bangladesh. The system is designed with a machine learning-based queue system that avoids wasteful waiting time for patients and improves coordination overall for doctors. Furthermore, a deep learning model was implemented for melanoma detection, which can enable the path for future more comprehensive disease detection via an online healthcare platform in Bangladesh.

Patients are able to schedule appointments online and see an estimate of waiting time in real time. The system has a melanoma detection feature using deep learning, where users can upload images of skin lesions for detection. More features such as, booking appointments, accessing medical reports and purchasing medicines has been implemented in the system with more features in mind for future. The main features of the system tackles both workflow inefficiency within clinics/hospitals and active health through detection of skin cancer.

HealthQ is the answer for the lapse of technology in digital healthcare platforms in Bangladesh. It can and will revolutionize the digital healthcare sector and bring forth the betterment to its entirety.

1.2 Motivation

Healthcare in Bangladesh faces many challenges which includes long patient waiting time, inefficient appointment management and limited access to digital disease detection opportunities. These problems not only is an inconvenience to the patient but can also lead to delayed diagnosis, delayed service to patients, worsening of medical conditions and in some cases preventable fatalities.

While digital health platforms exist in Bangladesh, it is still in the early stages of services such as tele medication or access to medical reports. There are no known platforms that utilizes intelligent technologies like machine learning to increase the efficiency of hospital workflows or support disease detection. This gap leaves both healthcare providers and patients with inefficient and incomplete solutions.

The idea of HealthQ comes from the need to address these inefficiencies and providing patients with more effective healthcare tools. The system does this by implementing AI powered queue management and image based disease detection (Melanoma). The project aims to reduce waiting times, doing so will increase efficiency in clinics/hosptials and maximize the output of our healthcare system can provide. It will also encourage better health management via disease detection.

Besides the improvements in efficient workflow, HealthQ is driven by a larger goal of demonstrating how modern technologies can be applied to real world healthcare challenges in Bangladesh. The project aims to set a foundation for smarter, more accessible and patient friendly healthcare systems in the future.

1.3 Objective

General Objective:

To design and develop an AI powered web-based healthcare platform that improves patient services. It will enhance healthcare delivery in Bangladesh through intelligent techonologies.

Specific Objective:

1. Reducing patient waiting times and unnecessary gathering by implementing a reinforcement learning based queue management system, improving the doctor-patient coordination.
2. Introducing disease detection via integrating a deep learning model for melanoma detection.

3. To develop an appointment scheduling module where patients can book and track appointments online.
4. To allow patients access to medical reports and health records through the platform.
5. Featuring additional features like medicine ordering and provide a scalable framework for future expansion
6. Implementing video consultation of patients and doctors and transcribing the video consultation for repeated use.
7. Comparing the current digital healthcare platform in Bangladesh and evaluating the performance and usability of the system

1.4 Methodology

HealthQ was developed using a modular design and hence various modules were modeled and built simultaneously. One of the team members was implementing machine learning and deep learning models development and the other one specialized in frontend and backend implementing. Specific development in technical fields was possible through this division as well as integration later on was possible.

Next.js was used to build the frontend of the system with the support of libraries like Framer Motion, used to create animations, TanStack Query, used to manage the state and NextAuth.js, used to safely authenticate users. Express.js was used to develop the backend, and MongoDB as a database, because of its flexibility and dynamic schema possibilities. The programming language was TypeScript that would be applied to both frontend and backend to improve the reliability and sustainability of codes.

The machine learning element is based on two aspects. To improve on patient scheduling and elimination of the queues, the first one was a reinforcement learning based queue prediction system (Q-learning) that helped to tell the right time to be scheduled to eliminate waiting queues. Second, an image classification model to detect melanoma was trained on the publicly available HAM10k dataset on deep learning. Many architectures, such as CNN, ResNet, Xception, EfficientNetB0, and InceptionV3 were checked, and the most successful one was chosen to be implemented.

The ML/DL modules have been integrated into a web platform through the usage of RESTful APIs. The final model to detect the melanoma will be under the API service, and the reinforcement learning system will work to refine the predictions with the possibility of periodic refreshment by cron jobs. The reinforcement learning agent was bootstrapped using a small custom dataset first to predict the queue.

The approach made sure that HealthQ will not only serve as a functional product as a web-based healthcare platform but also be smart with intelligent features driven by the capabilities of machine learning, and their highly scalable and modular structure will provide an extensible architecture in the future.

1.5 Project Outcome

The results presented in the HealthQ project prove not only the feasibility of such an endeavor, but also the usefulness of a platform with smart functionality, even in a small-scale environment.

Immediate Outcomes:

1. There was a development of a functional web-based health care system with appointment booking module, real-time queue estimation module, medical reports access module and medicine ordering.
2. A prototype queue management model is introduced, which is premised on reinforcement learning training and is able to minimize waiting time and maximise scheduling.
3. Web-based video consultation and audio extracting of appointments for patients to use for their own and using it as recorded version for repeated use.
4. With deep learning architectures, a melanoma detection module that was trained on the HAM10k dataset demonstrated acceptable performance, thereby allowing patients to upload skin lesions images to perform initial analysis using this module.

Technical Outcomes:

1. The workflow was successful because of the integration of machine learning and deep learning models, and scalable web platform using RESTful APIs.
2. Comparison of several different deep learning models (CNN, ResNet, Xception, EfficientNetB0, InceptionV3) and implementation of the one that provides the best performance in detecting melanoma.
3. Proved that reinforcement learning can be used to optimize queues in healthcare field which gives a stepping stone to continuing to make it better.

Practical Outcomes:

1. Healthcare services to patients include more intelligent services such as the provision of shorter waiting time, the availability of booking appointments online, and support in the detection of disease in the early stages.
2. It enables doctors and healthcare service providers to have better patient flow and a lesser administrative overhead.
3. The project indicates the possible solutions to inefficiencies in developing economies such as Bangladesh using intelligent healthcare platforms.

Future Outcomes:

1. Such a development in the identification of diseases should include expansion of those modules beyond melanoma so that it captures other conditions.

2. Further application to hospitals and clinics in Bangladesh that can potentially revamp patient experience and health care management.
3. Ensuring that HealthQ will be a scalable, modular system and may add new AI-powered medical services to it.

1.6 Organization of the Report

The report can be structured in six chapters to have a structured outlook of the project covering problem identification, implementation and evaluation of the project.

In Chapter 1 – Introduction, This section gives out the detail of the project including the motivation, objectives, brief description of the approach, the expected outcome, and a roadmap of the report. In chapter 2 (Background) it contains the requisite background study including reviewing the related literature, other related applications, and gap analysis to bring out the necessity of conducting this project. In Chapter 3 (Requirement Analysis and Design Specification), the section gives the general description of the system design and method followed, like the functional and the non-functional requirements, the architecture of the system, the data flow diagrams and the ui design. It also provides the detailed methodology that was undertaken and the project plan concerning task allocation. In Chapter 4 (Implementation and Results) it Describes the technical environment setup, implementation procedures, the test procedures, evaluation metrics, and reports the results together with a comparison analysis. Chapter 5 (Engineering and Design Challenges and Standards) it describes how the project meets software, hardware and communication standards, explain the design problems and analyzes ethical, social and environmental repercussions of the project. It also incorporates the task management and financial analysis, and charts the project against intricate engineering issues, types of problems, and engineering engagements. Chapter 6 (Conclusion) it will discuss affects a conclusion by summing up the work conducted, signs the limitations of the project and recommends potentially possible future work.

Chapter 2

Background

The chapter gives the background knowledge that one requires in order to comprehend the project. It comprises a literature search, analysis of similar applications, and the analysis of gaps to identify the drawbacks of current options and set the necessity of the given system.

2.1 Introduction

Digital platforms are becoming a common system in health care systems worldwide aimed at enhancing patient services, efficiency, and access to medical resources. The implementation of such technologies in developing countries such as Bangladesh is at an initial stage, and most available platforms provide only fundamental services, like teleconsultations or access to medical reports. Issues like patient queuing, poor scheduling and inaccessibility to early disease screening are still major problems.

The most recent innovations concerning machine learning and deep learning have demonstrated a tremendous potential in revolutionizing healthcare. The aforementioned technologies have already been utilized efficiently in a wide range of areas, such as medical image processing, medical prognosis, and operational optimization. Specifically, deep learning applications in dermatology have left lasting impacts on patient outcomes as conditions like melanoma could be detected in its early stages.

This chapter offers the background information that is required to understand the proposed project. It starts with an overview of pertinent literature on the detection of melanomas using deep learning. It then discusses the analogous use in digital healthcare and finds the gaps that exist, thus indicating the drive and the need to create the proposed system.

2.2 Literature Review

The digital health industry experienced incredible growth last year and is now expected to grow by over 30.2% annually, reaching a whopping $657.7 billion by 2026 [1]. While machine learning has promise for early disease diagnosis, it is still not widely used in hospitals. Fears about privacy, strict rules and a need to verify data many times have prevented more widespread use. Furthermore

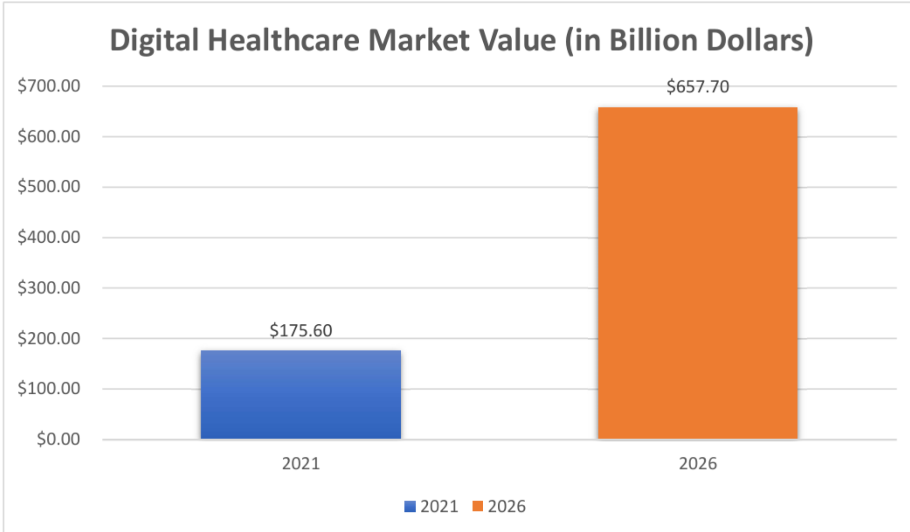

Fig 2.1: Digital Healthcare Market Growth

Nonetheless, the introduction of the machine learning (ML) into clinical practice is in a relatively comatose state. Recent surveys indicate that 12-25 percent of hospitals have implemented machine-learning based tools to detect diseases early [2], with most hospitals using ML to detect diseases being larger and armed with more resources. Factors that limit their broader usage consist of issues relating to patient information confidentiality, a large-scale check of algorithms, and regulatory restrictions [3]. Nevertheless, ML still holds a lot of promise in sectors like cancer screening, dermatology, and predictive analytics, which shows that soon, it might change the accuracy and efficiency in diagnosing illnesses and determining the fate of patients. Another Et al.

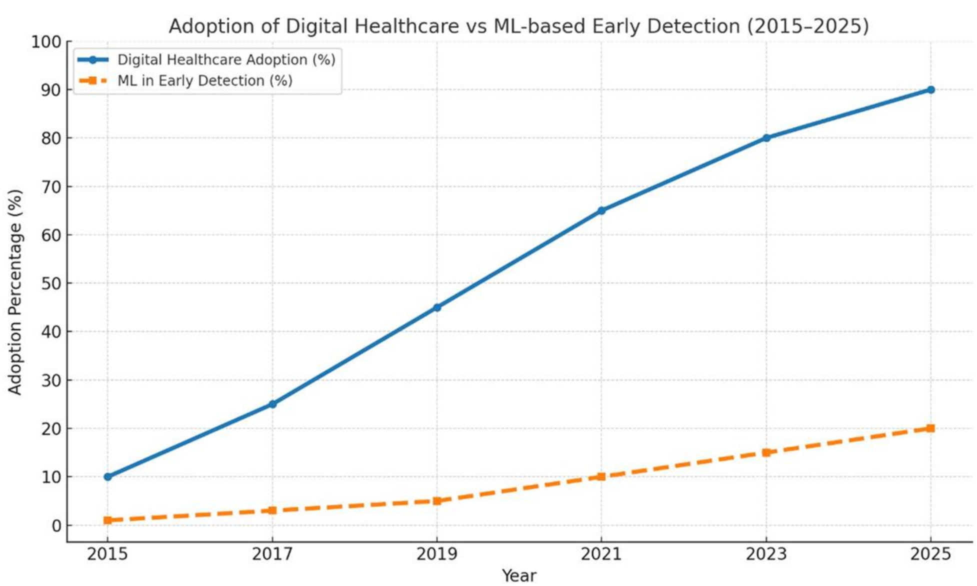

Fig 2.2 Adoption of Digital Healthcare vs ML-Based Early Detection

Bărcanescu gives an overview of AI use in digital healthcare systems, demonstrating that combining machine learning and deep learning is the most prominent enhancement of clinical decision support, diagnosis precision, and hospital workflow optimization [4]. The review highlights that although the modules enabled by AI can facilitate triage, signals abnormalities, and optimize processing, its true potential would be dependent on scaled deployment patterns and infrastructure-readiness, which has an increasing demand in the developing areas where data, compute, and interoperability limitations remain. Bhavnani et al. review validation systems in digital health solutions and believe that the platforms have shown growth in demonstrating an increase in accessibility and patient engagement though measures, but moves at a rate unable to keep up with rigorous evaluation [5]. The authors call on standard procedures, prospective reviews, and optimal reporting to ensure reliability and safety. They point to the continuing lacks in integrating predictive analytics and intelligent automation into routine care and point to the lack of strong validation in decelerating clinical trust and scaling adoption. The review literature on eHealth initiatives in Bangladesh suggests that the existing eco-system of telehealth is structured to revolve around the concept of teleconsultation and access to electronic reports, and there is little penetration of complex intelligent functionalities [6]. The analysis of unmet needs reveals queue optimization, predictive diagnostics, and patient-centered AI tools to be high priority when it comes to carry out the next stages of platform development. The future, they intimate, will depend on the work of improving infrastructure, consolidating data governance, and developing the local technical capacity to scale AI into business. Cayirli and Veral summarize an early work on outpatient scheduling and queue, showing that predictive modeling and optimization reduce a realistic amount of waiting time and improve flow [7]. The review lists the mathematical and simulation-based methodologies and mentions that, in those times, the lack of ML/AI restricted the ability to adapt to variability in demand, service times and no-shows and thus, inspired more reactive and data-driven approaches. Gupta and Denton introduce appointment and capacity scheduling analytical models demonstrating that properly designed policy can collectively improve the patient waits at the same time increasing provider utilization [8]. Focusing more on uncertainty and operation restrictions, they propose data-driven optimization that can be updated in near-real-time more often to help predict and address queues. They have suggested future ML-enabled systems, which combine such prediction with actionable schedule decisions. Patrick, Puterman, and Queyranne discovered how dynamic programming and simulation make an impact on the multi-priority diagnosis scheduling because reinforcement-learning-like policies can be used to optimize the shortage resources and reduce the waiting periods [9]. Their findings are a pre-cursor to current ML practices that take into account system dynamics and incorporate patient priorities during queue management rather than just focus on prediction and control. One of the latest research papers referred to as Malignant Melanoma Classification Using Deep Learning in 2020 also used the Convolutional Neural Networks (CNN) with the use of the IEEE, Springer, and other datasets; however, it demonstrated less than an average of results with a training accuracy of 70 percent and a test accuracy of only 30 percent, though it was of value in future research studies [10]. Another more recent study in 2022 presented the MobileNetV2 network, a lightweight, CNN-based, model, trained on a 70/30 split of the datasets and with 85 percent accuracy, but task limitations were introduced because this model was lightweight [11]. One more contribution made through a publication called Modernizing Analytics for Melanoma with a Large-Scale Research Dataset was the provision of the accessible large-scale dataset to do melanoma-related research, which implies the significance of robust datasets to make models effective [12]. Later, in 2019, a CNN Melanoma Lesion Detection (CMLD) model was created and demonstrated 90%-achieved accuracy in normal datasets and approximately 83 percent in cross-platform datasets as a pioneering application [13]. In 2021, through an analysis of clinical data covering 2000 patients and 33000 images, it was through the use of large clinical data that interest in the analysis of melanoma was shown [14]. In 2018, a study was implemented with 293 images to research with 83 percent accuracy; its limitation was the limited size of the data [15]. They have also been researched on non-melanoma detection where the use of segmentation techniques particularized by ML were examined on histopathology datasets which proved to perform well, though were narrow in outlook by not including melanomas [16]. Also, there was a report made in Annals of Surgical Oncology in 2010 about an electronic tool that can detect the presence of melanoma using the AJCC database, at least this tool estimated survivability instead of early detection [17]. In a similar way, a study published in the American Journal of Pathology also offered good advice on how to use ML to report the pathology of cancer correctly; however, a drawback was the details given on reporting rather than detecting [18]. One of the 2016 reviews was devoted to several approaches to the detection of melanoma based on different datasets without particular emphasis on ML techniques [19]. There was another similar investigation in the same year focused on the conventional means of early diagnosis of melanoma based on clinical data but without ML integration [20]. An automatic system of melanoma detection based on deep learning had a high-performance level by 2019 (95 percent accuracy and 92 percent dice coefficient in working with public datasets), but resource-intiveness was a limiting factor [21]. In 2012, survey of almost 300,000 cases (17,000 malignant and 283,000 benign melanoma) provided useful information since they did not provide data on dermatoscopy and digital monitoring [22]. Early Detection of Melanoma of 2000 mentioned ML applications but was scarce [23]. A subsequent 2009 study adopted nevi detection strategies using conventional methods and thus it paved a new way though it was flawed [24]. In 2018, a research study presented several ML techniques that can be used to identify melanoma but gave limitations because the methods use public/private data and require considerable computing resources [25]. A comparative study taken in 2019 compared several deep learning architectures using the ISIC datasets, reporting good performance but showing high resource demands, specifically very large computational requirements [26]. On the same note, Vocaturo et al. utilized ML classifiers of image processing (SVMs) and came up with high accuracy values based on huge training and validation sets [27]. This is another study aimed at detecting the stage of melanoma cancer on the basis of lesion images with 85 percent accuracy achieved but with a small amount of data [28]. Bigger data were demonstrated to enhance the predictive accuracy beyond 90 per cent, but obstacles of over-representation and explainability remained [29]. Other researchers did study recent challenges in ML and reached an accuracy of

75% using decision trees, although they reported that they would have preferred larger datasets [30]. The others researched future screening approaches [31], and then hybrid feature fusion models, when tested with various datasets of skin lesions, could reach 92% accuracy in case studies but complications encountered in their clinical implementation [32]. More innovative forms of detection (spectroscopy), had 94% accuracy but were restricted by specialized equipment, which did not encourage use [33]. SVM models as well gave an accuracy of around 90 percent with image processing and experienced overfitting [34], and when tested with various skin lesions data 87 percent of accuracy was obtained [35]. The studies linked to early detection reached 80 percent accuracy using stage-based melanoma and non- melanoma lesion data sets [36]. There were mobile applications with an 85 percent accuracy in detecting melanoma, but these were limited by the need to contain the system computing power [37]. The other methods like interpretable ML with multi-label prototyping scored 88% [38]. Lastly, Jones and colleagues showed that AI-based detection of melanoma would be at 92% accuracies using different algorithms, but such did not translate well to generalization due to the absence of consistent data and differing clinical contexts [39].

Table 2.1: Summary of Literature Reviewed

Author(s)	**Year**	**Methodology**	**Key Findings**
Bărcanescu, E.D.	2021	Survey of AI applications in digital healthcare platforms	ML/DL integration improves clinical decision support, diagnosis accuracy, and hospital workflow. Highlights need for scalable AI in developing regions.
Bhavnani, S.P., et al.	2020	Framework analysis of digital healthcare solutions	Digital health platforms improve accessibility and patient engagement but require rigorous validation. Gaps in predictive analytics and intelligent automation remain.
Islam, S.M.S., et al.	2020	Case study and review of eHealth initiatives in Bangladesh	Teleconsultation and report access dominate; gaps in queue optimization, predictive diagnostics, and patient-centered AI tools identified.
Cayirli, T., & Veral, E.	2003	Review of outpatient scheduling and queue management models	Predictive modeling and optimization reduce patient waiting time. Lack of ML/AI adoption at the time limited adaptability.
Gupta, D., & Denton, B.	2008	Analytical models for appointment and queue management	Efficient scheduling reduces patient waiting time and increases provider utilization. Data-driven optimization recommended for real-time predictions.

Patrick, J., Puterman, M.L., & Queyranne, M.	2008	Dynamic programming and simulation for multi-priority patient scheduling	Reinforcement-learning-like approaches improve resource use and reduce waiting times. Foundation for modern ML-driven queue prediction.
Naeem, A., Farooq, M.S., Khelifi, A., Abid, A.	2020	CNN-based melanoma classification using public datasets	Training accuracy 70%, testing 30%. Useful for future references despite low accuracy.
Indraswari, R., Rokhana, R., Herulambang, W.	2022	MobileNetV2 CNN classification	Achieved 85% accuracy with 70/30 train-test split. Lightweight design limits model refinement.
Richter, A.N., Khoshgoftaar, T.M.	2017	Large-scale dataset creation for melanoma analytics	Provides accessible datasets; emphasizes importance of large datasets for effective model performance.
Mukherjee, S., Adhikari, A., Roy, M.	2019	CNN model on cross-platform melanoma datasets	Achieved 90% accuracy on single-source, ~83% on cross-platform datasets. Innovative approach in the field.
Rotemberg, V., et al.	2021	Clinical dataset of 2000 patients and 33,000 images	Utilization of large clinical datasets highlights advantage in melanoma detection.
Auslander, N., et al.	2018	AUC analysis on 293 images	Achieved 83% accuracy; small dataset limits generalization.
Thomas, S.M., et al.	2021	ML-based segmentation for non-melanoma histopathology	Accurate segmentation, but limited focus outside melanoma detection.
Soong, S.J., et al.	2010	Electronic prediction tool using AJCC melanoma database	Estimated survivability rather than enabling early detection.
Scolyer, R.A., et al.	2013	Recommendations for pathology reporting using ML	Focused on accurate reporting, not detection.
Leachman, S.A., et al.	2016	Review of various melanoma detection methods	Lacked specific focus on ML models.
Tsao, H., et al.	2015	Review of traditional early detection methods	Utilized clinical datasets; no ML integration.
Adegun, A.A., Viriri, S.	2019	Deep learning system for automatic melanoma detection	Achieved 95% accuracy and 92% dice coefficient; resource-intensive.
Argenziano, G., et al.	2012	Large-scale multicenter survey	Valuable insights on 300,000 cases; lacked dermatoscopy and digital monitoring data.

Weinstock, S.M.A.	2000	Early ML-based melanoma detection review	Limited in scope; early efforts in ML detection.
Goodson, A.G., Grossman, D.	2009	Strategies for early melanoma detection (Nevi focus)	Introduced new directions; relied on traditional methods.
Okur, E., Turkan, M.	2018	Survey on automated melanoma detection	Highlighted ML approaches; limited by dataset size and computational needs.
Kassani, S.H., Kassani, P.H.	2019	Comparative study of DL architectures	Analyzed multiple DL methods with ISIC datasets; resource-intensive.
Vocaturo, E., Perna, D., Zumpano, E.	2019	ML + image processing for melanoma detection	High accuracy with SVM classifiers; large datasets supported training.
Patil, R., Bellary, S.	2022	ML for melanoma stage detection	Achieved 85% accuracy; limited dataset reliance.
Bhatt, H., et al.	2023	Survey of ML techniques for melanoma	Larger datasets improved accuracy >90%; overfitting and interpretability issues.
González Cruz, C., et al.	2020	Decision tree-based melanoma detection	Achieved 75% accuracy; larger datasets required.
Safran, T., et al.	2018	ML for future melanoma screening	Emphasized prospective screening strategies.
Rahman, M.M., et al.	2022	Hybrid feature fusion with ML	Achieved 92% accuracy; complex model may challenge clinical deployment.
Li, L., et al.	2014	Spectroscopic system + ML	Achieved 94% accuracy; specialized equipment limits adoption.
Mustafa, S., Dauda, A.B., Dauda, M.	2017	Image processing + SVM	~90% accuracy; overfitting issues.
Alquran, H., et al.	2017	SVM on diverse skin lesion datasets	Achieved 87% accuracy.
Al-Mashhadani, M.I.A., et al.	2019	Early detection ML models	80% accuracy on stage-based melanoma and non-melanoma lesions.
Dai, X., et al.	2019	Mobile ML app for melanoma detection	Achieved 85% accuracy; limited by mobile computing power.
Hussaindeen, A., et al.	2022	Multi-label prototype-based interpretable ML	Achieved 88% accuracy.
Jones, O.T., et al.	2022	Systematic review of AI/ML for early detection	Achieved 92% accuracy across algorithms; dataset variability limits generalization.

2.2.1 Similar Applications

When coming up with HealthQ, it is vital to check the similar applications that exist on the healthcare sphere where the strengths, weaknesses and gaps can be identified. Many healthcare Web and mobile apps already offer services such as appointment booking, telemedicine and diagnostics, though few are dedicated to queuing management with predictive analytics. Other Applications featuring similar application Case Studies

DocTime

1. Description: DocTime [40] Bangladeshi heavily focussed telemedicine app allowing patients to book online appointments as well as access general practitioners and specialists.
2. Methodological contribution: Offers remote consultations using mixed-media system and does not support real-time queue optimization or machine learning to predict service time.

The Popular Diagnostic Center App

1. Description: Popular Diagnostic Center App [41] provides diagnosis test reservation, health documentation, and bookings with doctors.
2. Methodological Contribution: An excellent level of integration with diagnostic services but has little to offer predictive queue management or smart scheduling.

Ibn Sina Healthcare App

1. Description: Ibn Sina Healthcare App [42] enables the reservation of consultations and diagnositic services in centres of Ibn Sina.
2. Methodological Contribution: Smooths the path to healthcare services but it does not use machine learning in the management of patient flow.

Practo (India)

1. Description: Practo [43] one of the most popular healthcare applications that offers appointment on the Fly, teleconsultations, pharmacy insides, and electronic medical records.
2. Methodological Strength: It could take advantage of its excellent ecosystem of digital health services, and it predicts the queue time in a generalized way instead of on a personalized basis.

Doctolib (Europe)

1. Description: Doctolib [44] a market leading European healthcare booking portal with online patient-doctor interaction management.
2. Methodological Contribution: Due to their widespread use and integration into the hospital processes, they are well known by their combination of large-scale adoption and integration into hospital workflows, however queue estimation is not an application of AI.

Maya (Bangladesh)

1. Description: Maya [45] specializes in delivering digital health and wellbeing services such as anonymous medical consultation over the chat.
2. Methodological Contribution: It is highly approachable and inclusive yet fails to include queue-like management and predictive diagnostics.

Appointment schedule and teleconsultation are the main schemes of most existing healthcare applications available but they have nothing to do with queue optimization. Systems like Practo and Doctolib also show some scalability and streamlining of workflows, however, they are not currently doing real-time queue prediction via machine learning, which is the central innovation of HealthQ. This gap presents the methodological advantage related to HealthQ in the combination between appointment booking, diagnostic integration, and smart queue management that creates a single platform.

2.3 Gap Analysis

The gap analysis shows that the current digital healthcare systems posses the features of appointments booking, teleconsultation, and diagnostic services, but none of them has advanced features of queuing time prediction based on AI technology, video consultation with the help of Google Meet, and synchronization of scheduling with Google Calendar. The innovative HealthQ system will fill this gap with a comprehensive intelligent patient-centered solution.

Table 2.2: Gap Analysis Summary Table

Features	DocTime	Popular Diagnostic	Ibn Sina Healthcare	Practo	Doctolib	Maya	Proposed System (HealthQ)
Online Appointment Booking	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Diagnostic Test Booking	No	Yes	Yes	Yes	Partial	No	No
Teleconsultation / Online Doctor Visit	Yes	No	No	Yes	Yes	Yes	Yes

Video Chat via Google Meet	No	No	No	No	No	No	Yes
Queue Time Prediction (AI/ML)	No	No	No	No	No	No	Yes
Google Calendar Event Integration	No	No	No	No	No	No	Yes
Electronic Health Records (EHR)	Partial	Partial	Partial	Yes	Yes	No	Yes
Patient Engageme nt / Chat Support	Partial	No	No	Yes	Yes	Yes	Yes
Accessibili ty in Developing Regions	Yes	Limited	Limited	Limited	Limited	Stron g	Strong
Transcribin g video consultatio n for repeated use	No	No	No	No	No	No	Yes

2.4 Summary

In this chapter, what has been done or studied up to date in the field of digital healthcare platforms and queue time prediction has been discussed. Studies were published on the skyrocketing of digital health solutions and their promise to improve accessibility, the accuracy of the diagnosis, and workflow management but there are issues with their validation and application of AI. In the case studies in Bangladesh, the intelligent features are found not to have been used extensively, so there was a need to introduce predictive and patient-centred features. Less investigations into queue management issues were based on studies on queue scheduling and optimisation programs and more recently, dynamic programing and learning-based schemes have been applied to the waiting times. The examination of the available healthcare applications has revealed the lack of features powered by AI (the queue anticipation, video consultation, and calendar), and the identified gap can be addressed with the HealthQ developed and proposed system.

Chapter 3

Research Methodology

The procedures, tools and techniques which were used in the development of the proposed HealthQ system are outlined in this chapter. It outlines how the research is to take place, the instruments used to collect data, the architecture of the system, and how the approaches will attain the project aims.

3.1 Requirement Analysis & Design Specification

3.1.1 Overview

The requirements analysis and design specification forms the basis of the development of the HealthQ system. This section gives an overview of the identified essential needs on the basis of user research, cases studies of existing platforms and analysis of gaps in care provision. First of all, it is important to make sure that the system focuses on both functional and non-functional requirements, including the accurate scheduling of appointments, the prediction of queue times, and video consultations, along with reliability, security, and simplicity of use. According to these requirements, the design specification provides the structural plan of the system, its architecture, database design, and third-party tools such as Google Meet and Google Calendar. This will make sure the given solution is feasible technically and fulfils the expectation of the user.

3.1.2 System Design

HealthQ is in development as a modular web-app which is accessible via the user's web-browser, for desktop and mobile use. The system is comprised of some major modules: User Interface, Backend API, Database, AI/ML Engine and External Integrations. User Interface : Interactions between patient and doctor while booking the appointment, joining the queue, checking the report, buying the medicine. The front end interacts with the Backend API, which is the core of the platform, that is in charge of receiving requests, performing business logic and performing authentication and authorization (using NextAuth) with role-based access control as a security mechanism for patients and doctors.

AI/ML Engine is a separate service in the same tier but looking at patient data to estimate queues for wait times and early disease diagnosis suggestions. This engine is asynchronous with the backend, which means you can process in real-time without blocking your application. Database records patient information, appointment, queue status, AI results and user input preference safely stored in the database, encrypted storage and privacy compliant. It has been combined with other systems (Google Meet/ Google calendar to schedule the medical consultation in an online way, the payment system, the notification system,), to provide optional functionalities that enhance the user experience, namely: the video call consultation and booking. The use of event-based communication comes in to support real time statues of a queue or appointment confirmation to ensure the stability of operations.

The system will incorporate a Video Consultation module in it under which patients and doctors will be able to hold virtual meetings. Recordings of the sessions are made and the audio is mined and sent to an AI non-speech transcription service (Whisper-1) [46] where accurate text transcripts are generated. This module will watch with the backend APIs and database such that the transcripts might be stored in a secure point and should be retrieved in the patient and doctor dashboard. The benefit of the modular architecture is that the transcription problem does not affect other system components.

Lastly, to make sure that the whole system can be enhanced with new features or with third party services that have to be used in the future, the design of the system is modular, secured, scalable and can be integrated with the AI-driven healthcare analytics, providing a strong foundation for expansions.

HealthQ is a web app designed as a module that can be viewed through both desktop browsers and mobile browsers and built around large modules such as the User Interface, Backend API, Database, AI/ML Engine, and External Integrations. The frontend enables patients and doctors to manage appointments, queues, reports, and purchasing of medicines, whereas the backend fulfills business logic, authentication and role based access control with NextAuth. The AI/ML Engine is an asynchronous queue predictive, early disease detection system with all data regarding patients and the system being safely stored in the database. Online consultations can be arranged using external tools such as Google Meet and Calendar, and a Video Consultation module records the consultations, extracts audio and uses Whisper-1 to perform transcription accurately, the transcripts are stored on a secure dashboard so they can be accessed when needed. The safe, scalable and modular design makes it stable into the system, allows easy integration of next generation features and interaction between AI, back-end and external services.

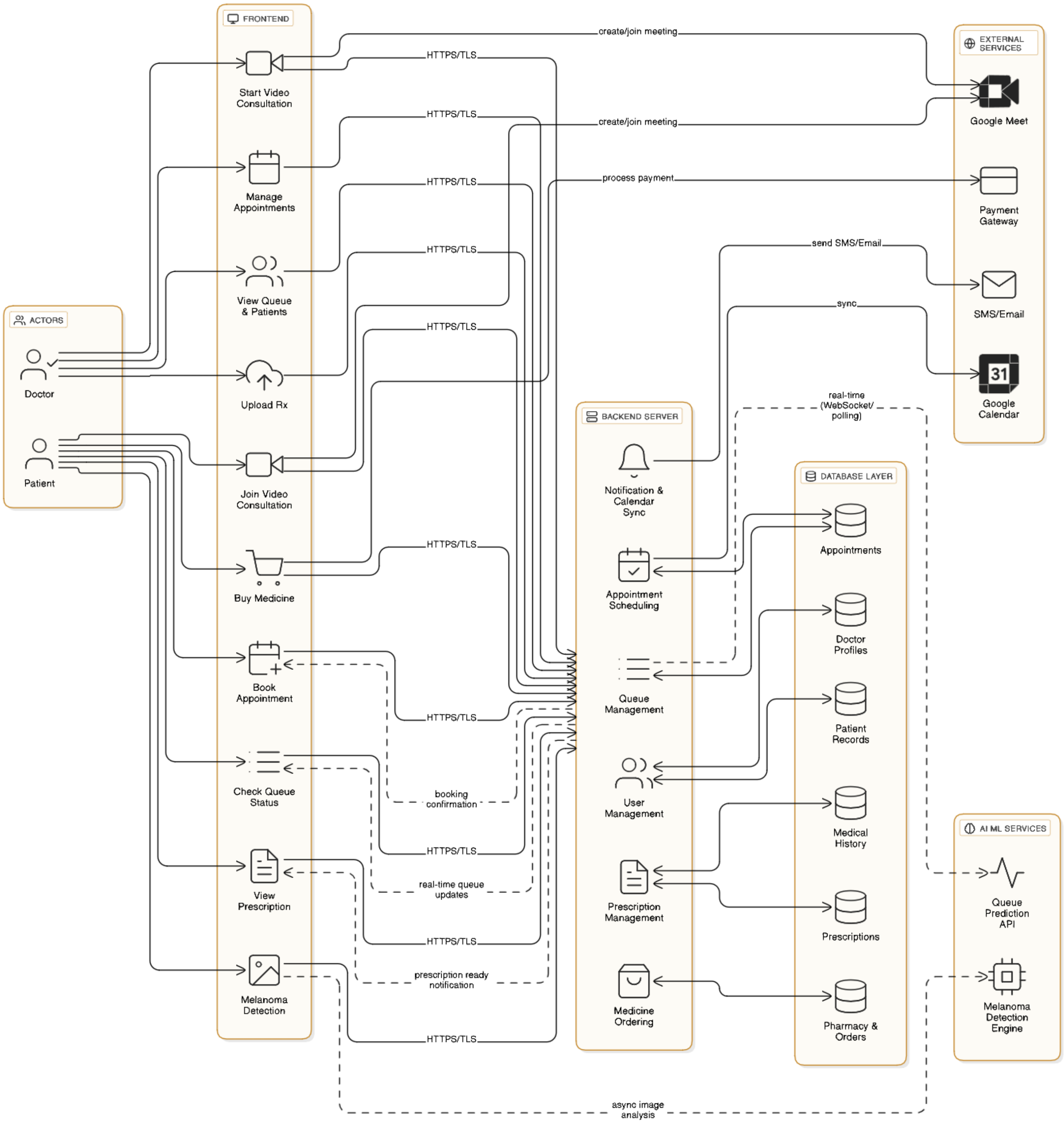

Fig 3.1: System Architecture of HealthQ [49]

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements

1. **User Control**: Patients, doctors and administrators should be able to register, log in and out with the system. Individual users will be role-based to allow access to relevant information only to the authorized persons. The user should also have the option of updating their profile and contact details.
2. **Appointment Management**: Patients are supposed to have an ability to book, cancel, or reschedule appointments with doctors. Doctors may garden view, confirm or modify appointments. Any changes in the appointments must be communicated on a real time basis to both parties by the system.
3. **Queue Management**: The system ought to show a queue detail in real-time for every doctor or service. Patients can view queue position and approximate ques. Text messages must warn patients when they are about to be called up or when there is a delay
4. **AI/ML Predictions**: The system would automatically include machine learning techniques to identify early stages of the disease using data collected about the patient and predict queue times so that it can better plan resources. Individual I results are required to be displayed in a manner in a manner that will help doctors make judgments without displacing human judgment.
5. **Access to Medical Records**: Patients should have the ability to view, upload and manage their medical records, which includes test results and reports, in a secure manner. Patient records should only be accessed by the doctors in case of proper diagnosis and treatment.
6. **Video Consultations**: The system must be able to enable patients and physicians to do tele-consulting using Google Meet. The users should be able to schedule, join, and leave video calls smoothly in the platform.
7. **Transcribing Video Consultation**: Transcribing video consultation via Whisper-1 and OpenAI axios.
8. **Integration with Calendar**: bookings made through the system must also integrate with a Google Calendar, reminding patients and doctors of upcoming appointments, and letting them manage their time accordingly.
9. **Purchasing Medicines**: The patients should be in a position to search drugs available, order and trace delivery progress. The system needs to offer alerts on medicine availability, scripts and medicine refill reminders.
10. **Notifications and Alerts**: The system will need to send timely notifications through email or SMS confirming appointments, cancelling appointments, queue updates, AI prediction results and other events of interest to the system.

Nonfunctional Requirements

Table 3.1: Nonfunctional Requirements for HealthQ

Requirement ID	Description	Type
NFR-01	System response time ≤ 3 seconds under normal load	Performance
NFR-02	Uptime ≥ 99% for critical services (appointments, notifications)	Reliability
NFR-03	Support for increasing users, doctors, and AI modules without major redesign	Scalability
NFR-04	Encrypt sensitive data, secure authentication, role-based access	Security
NFR-05	Intuitive UI/UX, mobile-responsive, accessible to all users	Usability
NFR-06	Modular architecture allowing updates to AI models, APIs, or UI independently	Maintainability
NFR-07	Compatible with modern browsers and integrates with Google Meet/Calendar	Compatibility

3.1.4 Data Flow Diagram

The Data Flow Diagram (DFD) is a visualizing of flow of information in the HealthQ system between users of the system, processes and services provided in the system as well as data stores. It gives the visual representation of the functionality of the computer system, including connections, like appointment booking, queues, AI-based predictions, video consultations, and prescription sale. The DFD has been useful in determining the overall structure of the system and makes the system easy to comprehend in regards to how the inputs can be processed to come up with the desired outputs.

After the overview, the data flow of the HealthQ system will be represented using various levels of the Data Flow Diagrams so as to give an insight into the working of the system. The top level DFD-0 (Context Diagram) shows the system as a single process and outlines its interactions with the outside world: patients, doctors, AI/ML engine, Google Meet, Google Calendar, and pharmacy APIs. The following high-level diagram shows the primary data flows into and out of the system, but in simplified view such as the overall structure. The next level, DFD-1, subdivides the system into a more detailed set of processes that corresponds to the functional requirements. As an example, the booking and queue management process displays how the requests of patients are managed, changed, and transferred to the AI engine to calculate the wait time. Video consultation process illustrates the flow of data used in initiating, conducting and storing consultations via Google meet. Likewise, the medicine buying process exhibits the flow between a patient order and a confirmation of the pharmacy. The AI/ML prediction module is also drawn to illustrate how patient data enters the AI engine and the prediction is returned to the system. Complex modules involving additional details can have other DFD-2 diagrams inserted as subsidiary process descriptions to them to accord them a more detailed insight and description. This sort of engineering approach will result in all functional requirements being captured, and the flow of information throughout the system being well documented.

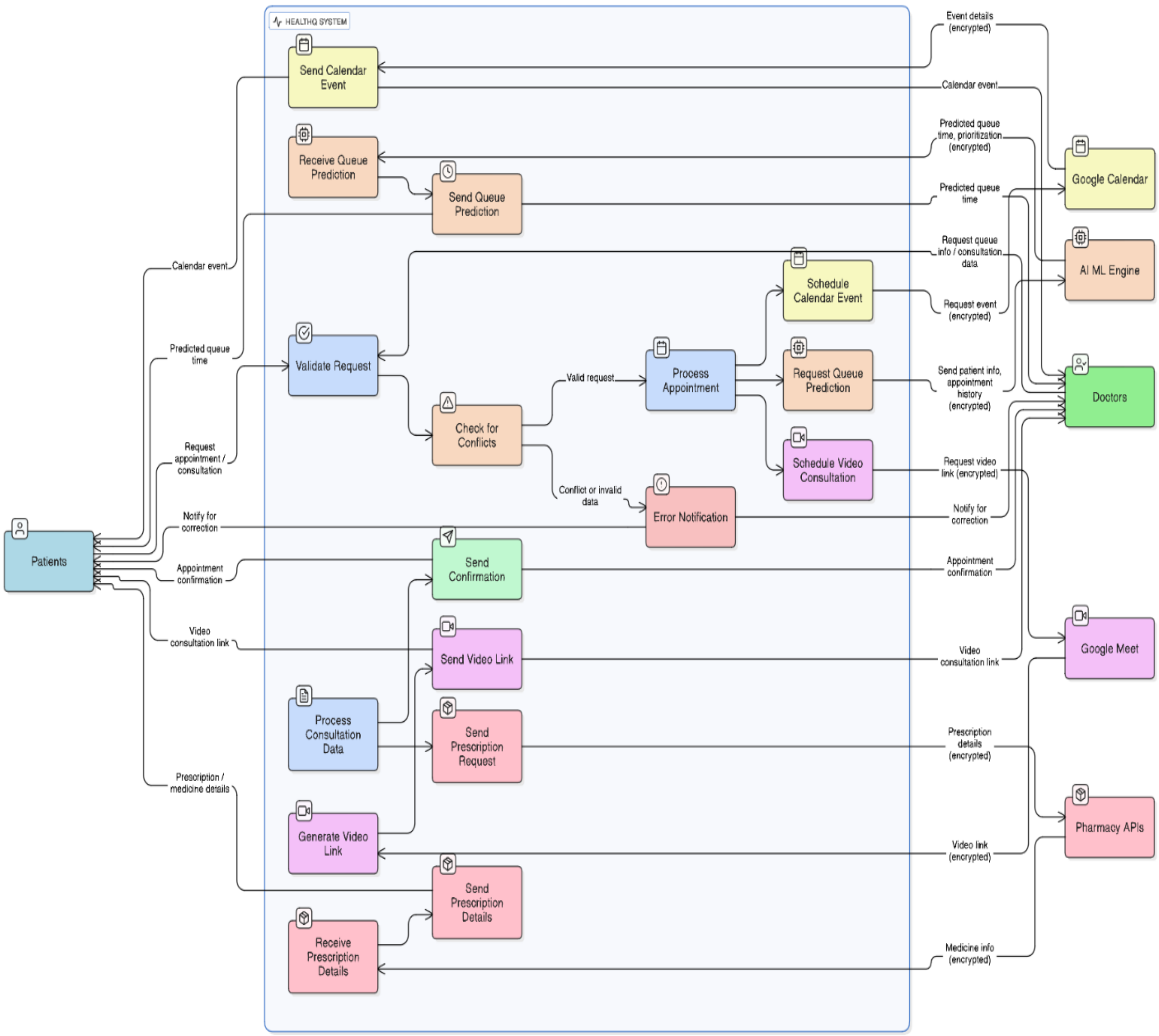

Fig 3.2: DFD-0 of HealthQ

The context diagram of HealthQ gives an overview of the system at a high level, which gives a view of the system as a single process that interacts with other external entities. Important entities are Patients, Doctors, a AI/ML Engine, Google Meet, Google Calendar and Pharmacy APIs. It depicts the key pieces of information flow including appointment requests, queue information, consultation, and video links, calendar events, and prescription information. This diagram focuses on the external interactions of the system and isolates the AI/ML Engine in order to distinguish its specialized role, and provides a clear big-picture view on how HealthQ interacts with users and the rest of the outside world.

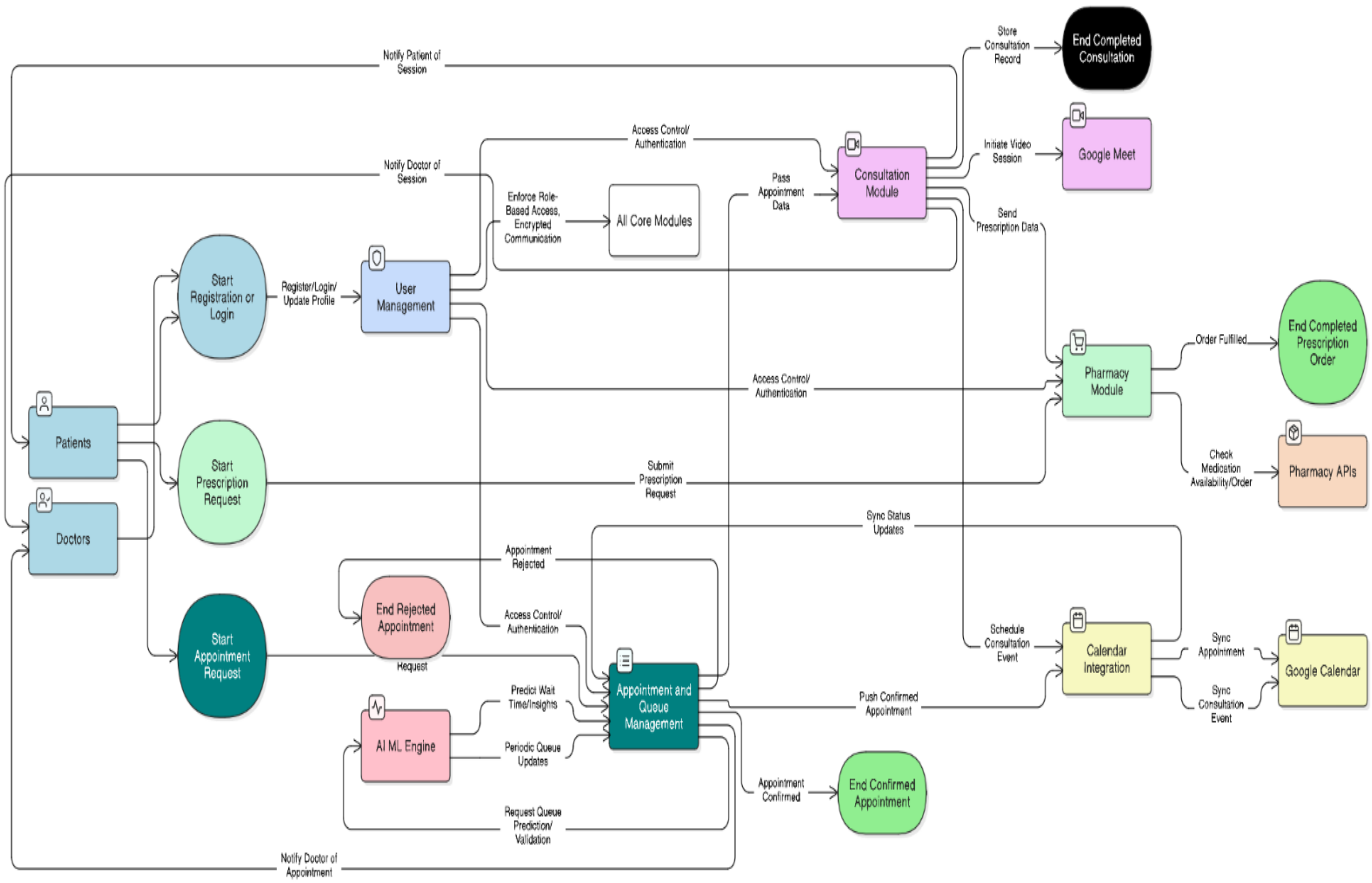

Fig 3.3: DFD-1 of HealthQ

The Level 1 Data Flow Diagram (DFD-1) of HealthQ summarizes the main modules and their interrelations according to the basic functions. It breakdowns the context-level diagram (DFD-0) of the interaction between the data items that flow between the patients, doctors, the AI/ML engine, and video consultations modules.

In this figure, patients place appointment requests that are validated by the system and the processing of the request is done with queue predictions based on the engine of the AI/ML. Doctors have access to the patient data, hold consultations through Google Meet and continuously update patient records. Video consultation module manages the initiation, connection, management and storage of consultation data. The focus of FD-1 on high-level interaction between modules as well as the lack of internal sub-processes will make it clear how the key modules of HealthQ are communicating and focus on data flow associated with the appointment scheduling, AI predictions, and real-time video consultations.

At this level of Data Flow Diagram, what it consists of is to concentrate on the queue time prediction process in HealthQ system. Data used to train AI/ML engine when a patient joins the queue and requests an appointment or when a patient requests an appointment include service times of a doctor with his or her ID, specialty, and appointment type, and also includes historical data about such service times. This information is fed to the AI/ML engine to predict the expected wait time of the patient. This wait time can then be fed back into the patient interface so that the patient is aware of how to schedule.

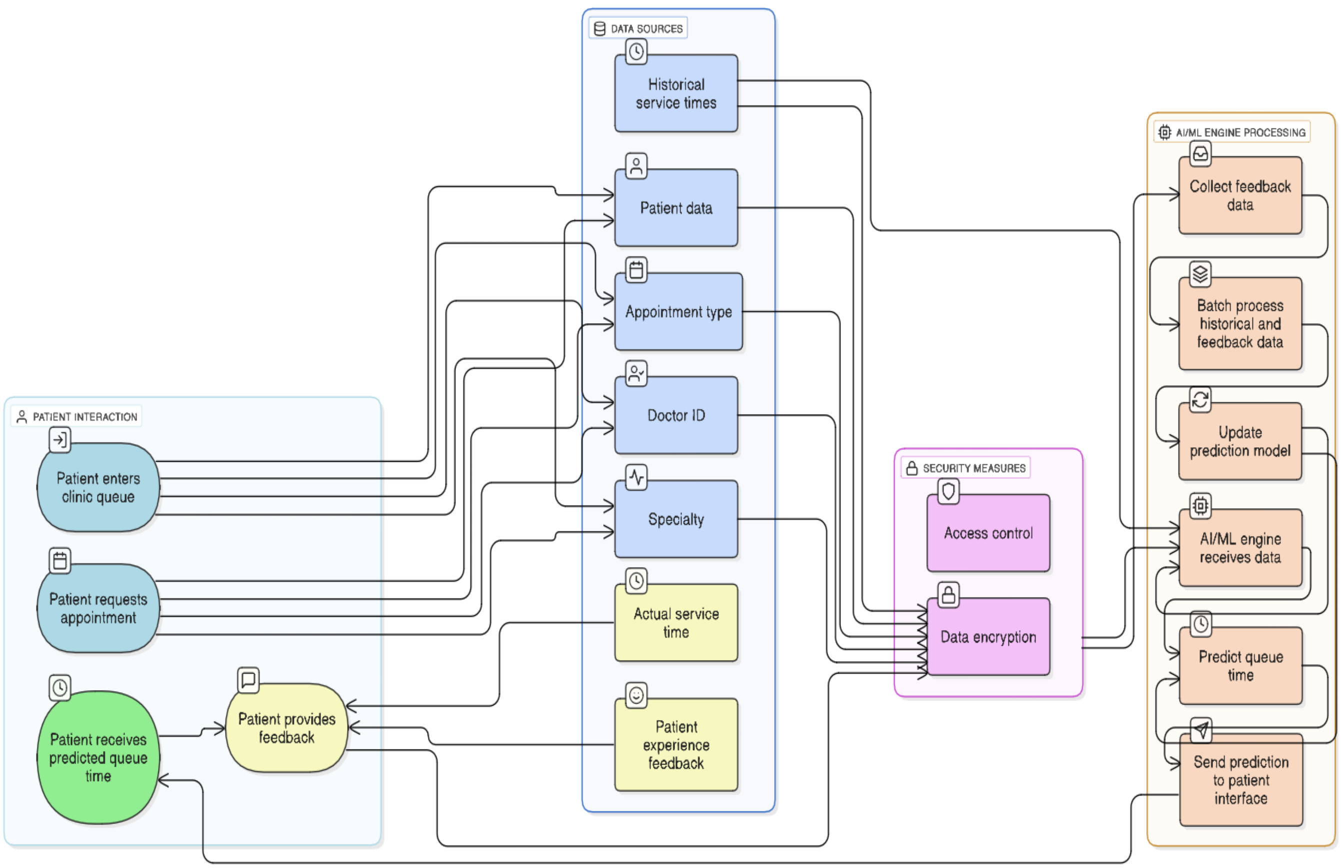

Fig 3.4: DFD-2 of HealthQ (Queue Time Prediction)

In addition, the informed response of the actual service times and patient experience after the attendance is also received and provided back to the AI/ML engine to update and enhance the prediction model over time and achieve a constantly increasing accuracy. There are time-based data flows where every interaction can be used to update the queue and there may also be the batch data that is historical and feedback data and this can be optimized through the batch processing. Security and privacy of data such as encoded information transfer systems of protocols and restricted data access is also presumed as it is the data of the patient.

3.1.5 UI Design

HealthQ User Interface (UI) design puts patient-friendly, intuitive, and accessible standards at the forefront with both patients and doctors. Rather than displaying all the screens in the system, this section highlights the most important ones that directly facilitate the key functions of the solution, which are bookings, queue status visualization, and access to the video consultation. By focusing on the following key aspects, the design focuses on clarity, usability, and efficiency without confusing the user with detailing that is unnecessary.

Patient Dashboard: The Patient Dashboard is the core program where a user can organize his or her healthcare endeavors in HealthQ. It shows an overview of the upcoming and past appointments with information like a doctor name, specialization, and consultation type. An important aspect of this dashboard is that the queue management was shown in a manner that allowed the patient to gauge his/her position in the queue, as well as the estimated time of waiting, work out by the AI engine. This real-time data can save uncertainty, as patients can plan how and when they approach the health center and conduct video consultations, and it will make the healthcare system more transparent and effective.

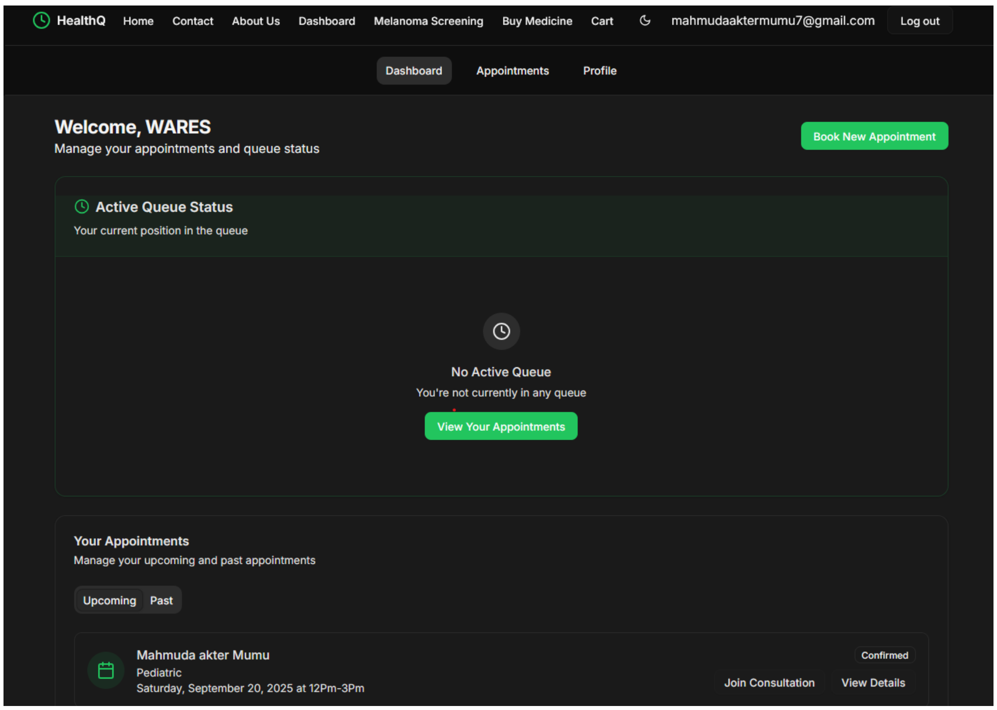

Fig 3.5: Patient Dashboard of HealthQ

Doctor's Dashboard: The Doctor Dashboard has been set up to simplify the daily work load of the healthcare practitioner who uses HealthQ. It shows a well-organized agenda of upcoming, and current visits, including patient names, purpose of visit, and type of visit (face-to-face or video). Doctors are also in a position to see the flow of the queue i.e. the order in which they have to see patients and even the time they are likely to consult patients. The dashboard is also compatible with video appointment services, such as Google Meet, and Google calendar synchronization! Having all the necessary information at an interface will reduce administrative workload and result in doctors spending more time diagnosing patients.

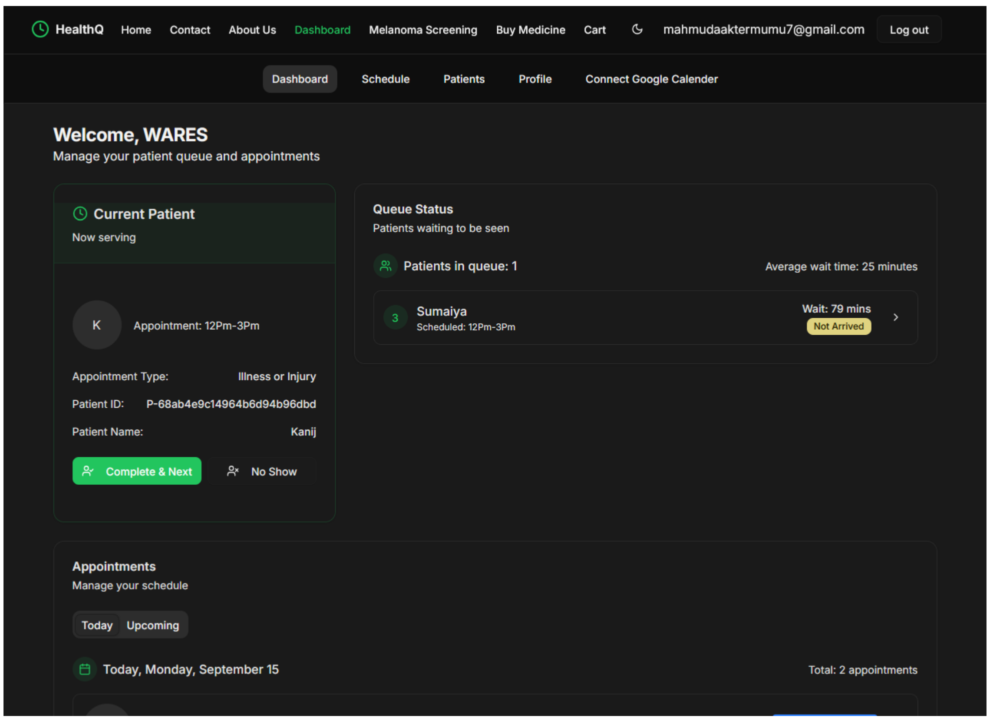

Fig 3.6: Doctor Dashboard of HealthQ

Buying Medicine: The Buying Medicine interface offers patients an easy mechanism of ordering prescribed drugs through the HealthQ platform. Once a consultation is done, the system creates an e-prescription that will be reviewed by the patient and can be used to place an order. The interface lists available medicines with the search option, description, price, availability status. Patients may add the medicines to the cart, choose quantities, and complete the purchase. The system also integrates with partner pharmacies which will allow home delivery or in-store pickup. This feature allows patients to access their medications faster as it is easier to buy the medication without having to travel to a brick-and-mortar drug store.

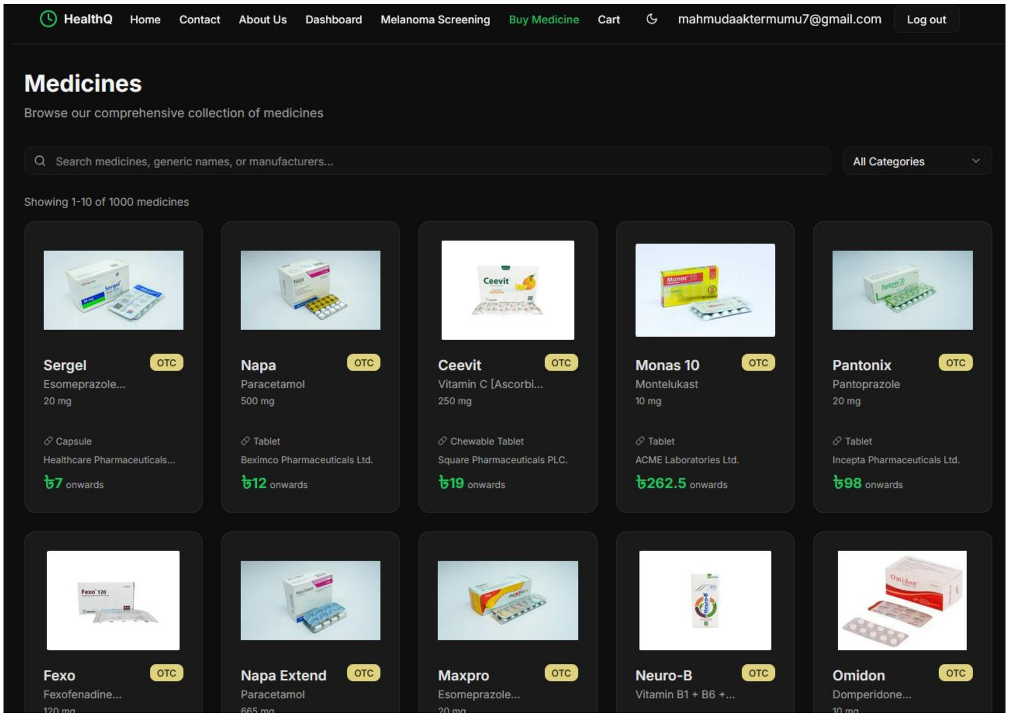

Fig 3.7: Medicines Purchase of HealthQ

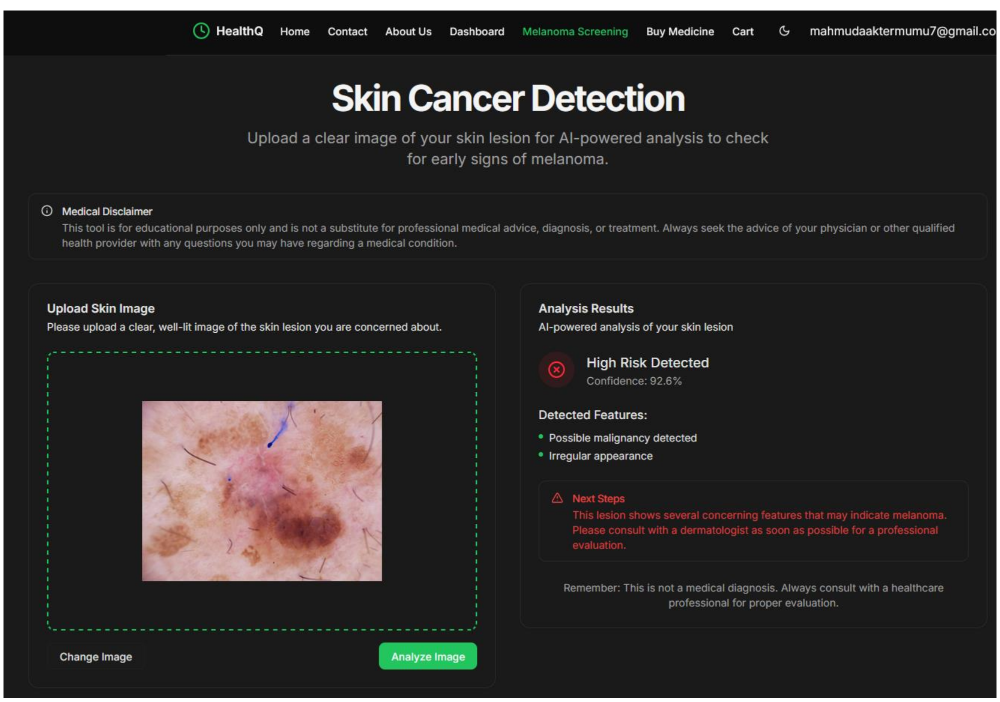

Fig 3.8: Melanoma Detection of HealthQ

Melanoma Detection: Melanoma Detection interface enables the patient to upload their skin lesion pictures to be analyzed by AI. The user interface is meant to be hassle-free and intuitive, with clear directions in taking or choosing a photo. After uploading an image, the system runs through it in the AI engine to provide the results in an easily comprehensible format as to whether the lesion is probably benign and safe to ignore, or whether it needs further medical treatment. A follow-up recommendation, appointment with a dermatologist, is also provided in the interface. This will guarantee fast access to quality information by patients without losing a user-friendly interface.

Audio Transcription: The Audio Transcription UI application part of HealthQ allows patients and physicians to have a convenient access point to the results of video consultations. After recording a session, it is transcribed with the help of Whisper-1, and a text is presented in a specific dashboard block. Vitals or other vital information like medical notes are pulled to the top so that important information is easy to review, and the entire transcript may be accessed at the bottom. Another characteristic of the UI is its inclusion of patient information as well as the transcript, thus allowing physicians to make timely decisions and care continuity. Its design makes it readable, clear and allows easy access to the pertinent information without congesting the main patient or doctor dashboards.

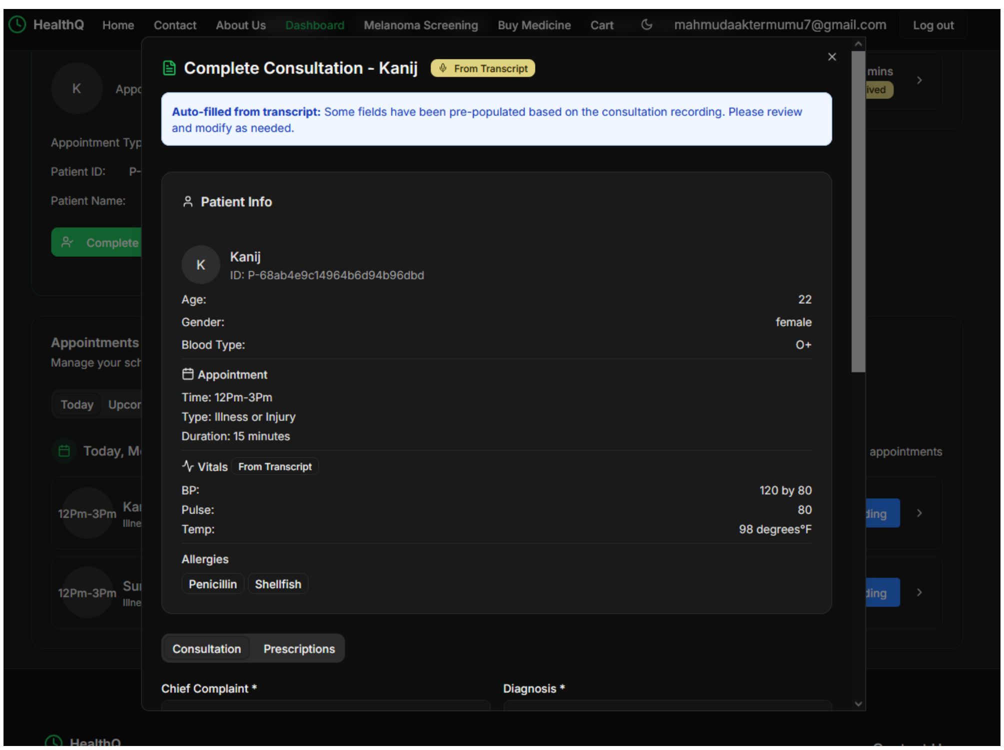

Fig 3.9: Audio Transcription Output of HealthQ

In a nutshell, the HealthQ user interface is intent on clarity, ease of navigation, and simplicity among all modes of access including patients, doctors, and administrators. Every screen and every functionality such as appointment view and queue monitoring, melanoma recognition and medication ordering are created to give users all the necessary information in the most convenient way possible. The UI will focus on accessibility, responsive design (on the web and mobile), and ease of use with an aim to improving user satisfaction and engagement.

3.2 Detailed Methodology and Design

In this section the detailed methodology and design decisions made during the development of HealthQ system are presented. It brings to fore the choice of technology stack, methods of queued time prediction, third party service integration methods, modular system construction, and AI/ML model choice. Development alternatives contemplated are also discussed and reasoning why it chose the latter provided. The methodology will make the system efficient, scalable, reliable and at the same time flexible to future upgrading.

3.2.1 Technology Stack Selection

HealthQ system is designed with a blend of frontend, backend, and machine learning technology in order to make it scalable, maintainable so that its connections with the AI models are efficient.

Frontend: The front end was created using Next.js which is a react based framework that supports server side rendering, static site generation, and integrating APIs in a seamless way. To make the UI even more pleasant, Framer Motion was used to implement both animations and transition, whereas TanStack Query was used to manage the state efficiently as well as perform the asynchronous data fetching. To increase the level of authentication, NextAuth.js was included to add numerous ways to log in, and JWT Decode was included to handle the tokens on the clients.

Backend: The backend codebase was written with the help of Express.js, a light-weighted framework developed on top of Node.js, to work on RESTful API and server-side codes. MongoDB was selected as the database since it has a flexible schema that will be able to store patient records, appointments and output of the AI models dynamically.

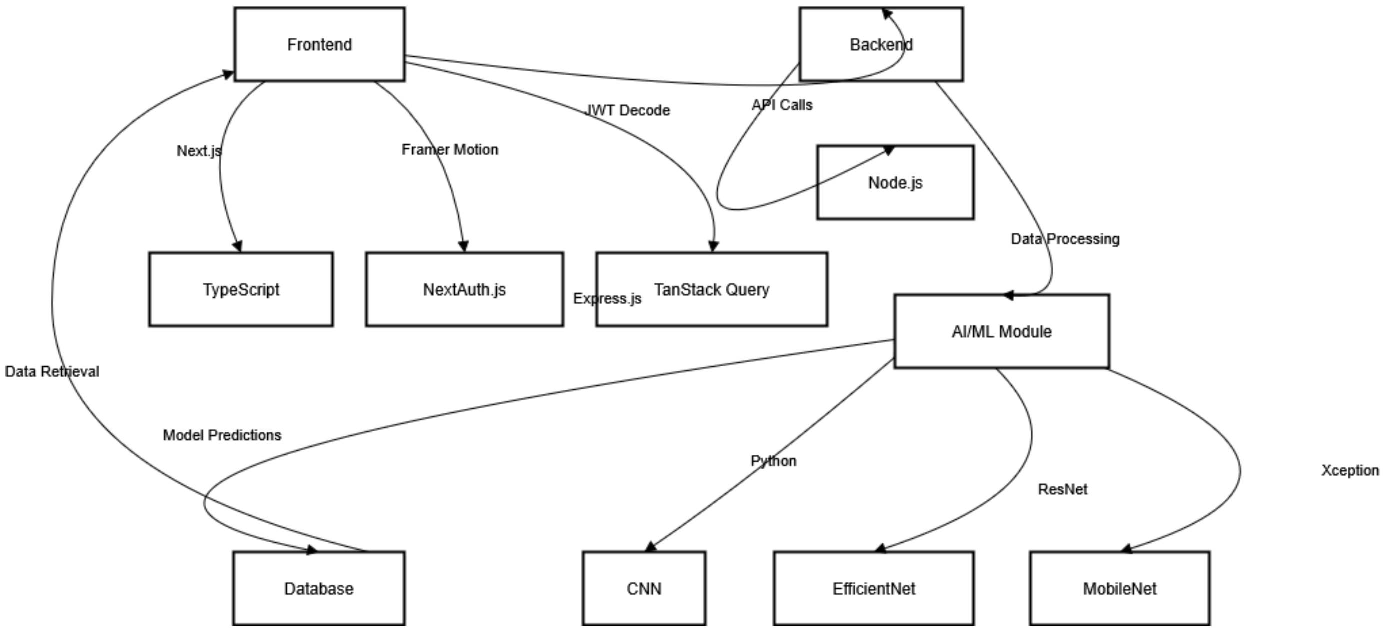

Fig 3.10: Technology Stack of HealthQ

Programming Languages: It was selected that both frontend and backend development was conducted using TypeScript since type declarations provide the benefit of stronger code reliability, while Python was selected to produce AI/ML models as it has great library support, and it is easy to experiment.

This combination of technologies has led to a system which is modular, maintainable and scalable enough to include any advances AI-driven feature without sacrificing performance.

3.2.2 Queue Prediction Model Selection

Various modeling methods in the design of the queue prediction framework of HealthQ were discussed in the design with the view of providing realistic and accurate patient waiting time prediction. First of all, the reinforcement learning was regarded as the first approach because of its capability to optimize the multi-variable dynamic systems. Nevertheless, the availability of real-time data is low, and it is too risky to overwhelm the system with generated data, so the initial implementation used a simpler, less intrusive approach.

A supervised regression model was applied initially using available sample data to initially predict baseline values of queue time. This strategy enabled the model to come up with solid predictions without overly introducing noise and bias using limited datasets. When actual data started to be generated in the course of actual operations, the system inculcated a reinforcement ability to learn in batches at the end of each and every day. This combination made the queue prediction model continually adjust itself using the observed data but still retain the stability of the supervised model.

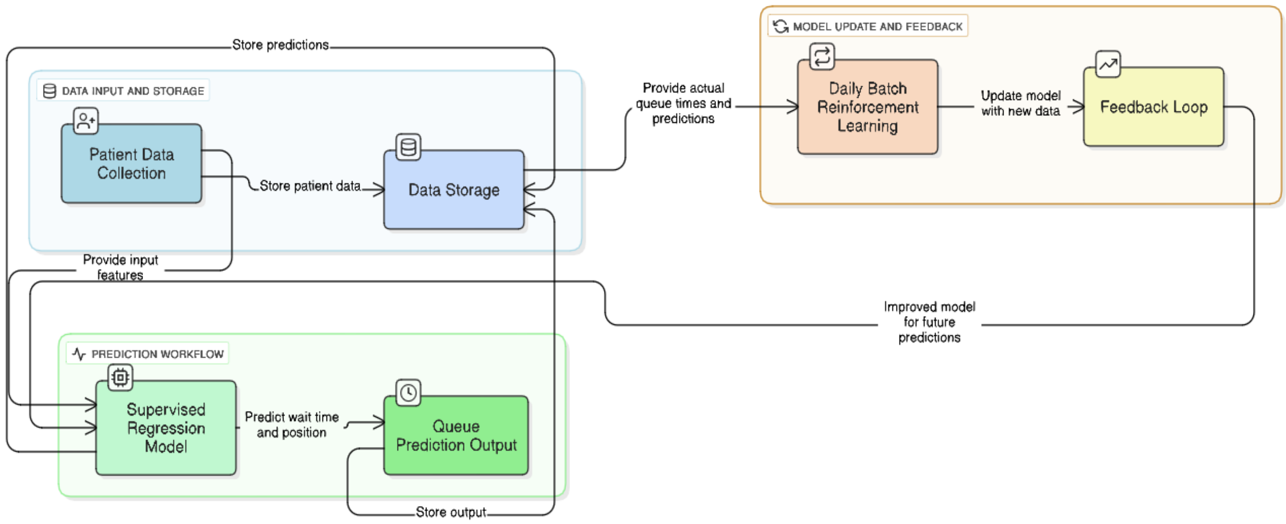

Fig 3.11: Queue Prediction Method of HealthQ

This approach was used to balance between precision, stability and flexibility, enabling HealthQ to suggest meaningful predictions to patients, and establishing a base on which to improve as additional data in real-time can be acquired.

3.2.3 Third-Party App Integration

To attain greater usefulness and provide an improved user experience, the HealthQ system is augmented with third party Alan Kay parallel sophistication, that is, rather than creating everything in-house we select additional functionality when porting carefully selected third party services. Such design would ensure a high level of reliability with less development cost and increased access to enterprise-level capabilities.

The most impressive among the integrational possibilities is that of Google video consultations that takes place through Google Meet. Instead of developing their own video conference system, and having to put a serious amount of effort into making it stable, encrypted and scalable, Google Meet is a service healthy enough to fit most of their needs. It has tried and tested infrastructure that ensures fast patient-doctor communication with a low latency and screen sharing capabilities and encrypted communication as this is required when health-related discussion occurs.

The system also interacts with Google calendar in order to schedule and synchronize appointments. Such integration will become possible to send automatic reminders, control the schedule of the doctors and patients without the need to make a number of calls and exclude the possibility of having the patients booked simultaneously. Using Google calendar, HealthQ does not need to create its patient/doctor syncing scheduling calendar and notification system, but it is able to give real time day-to-day patient/doctor coordination.

The authentication, session management includes the integration of NextAuth.js and OAuth providers along with the safe session management. This connection is combined with effective authentication measures (email, credentials, or third-party logins) without undermining privacy of sensitive healthcare data.

All of these integrations have been chosen because they do not take too much time to develop, are scalable and suport industry level security and privacy standards. The cumorous functionalities then shift to the trusted platforms that leaves HealthQ to focus on its core healthcare functionalities such as the queue forecast, melanoma diagnosis, and meaningful patient-doctor interaction.

3.2.4 AI/ML Model Selection

The HealthQ platform draws on Artificial Intelligence (AI) and Machine Learning (ML) to perform two important functions, namely queue time prediction and melanoma detection. The model selection has been done through a systematic process based on the size of the data set, performance measures and practical feasibility of the deployed model.

In the case of queue time prediction, the system used a supervised regression model based on structured information like demographic data of the patient, the specialization of the physician and the type of appointment. This strategy gave a starting point with which waiting times could be estimated fairly accurately. In order to increase the adaptability there was an addition of reinforcement learning (RL) element in order to update the batches in a day. Daily after the end of the working day, the model is adjusted using the real-life performance streams of queue times. This prevents the system overfitting a synthetic data and progressively growing by iteration.

To detect melanoma, we used the HAM10000 dataset (Human Against Machine with 10000 training images) [47], a set of dermatoscopic images of various pigmented lesions of the skin. To overcome the drastic skew in the proportions between the seven diagnostic classes, the data were reclassified into a benign versus malignant broad category. This dichotomous classification enhanced model concentration and derivativeness. After such restructuring however there was still some imbalance.

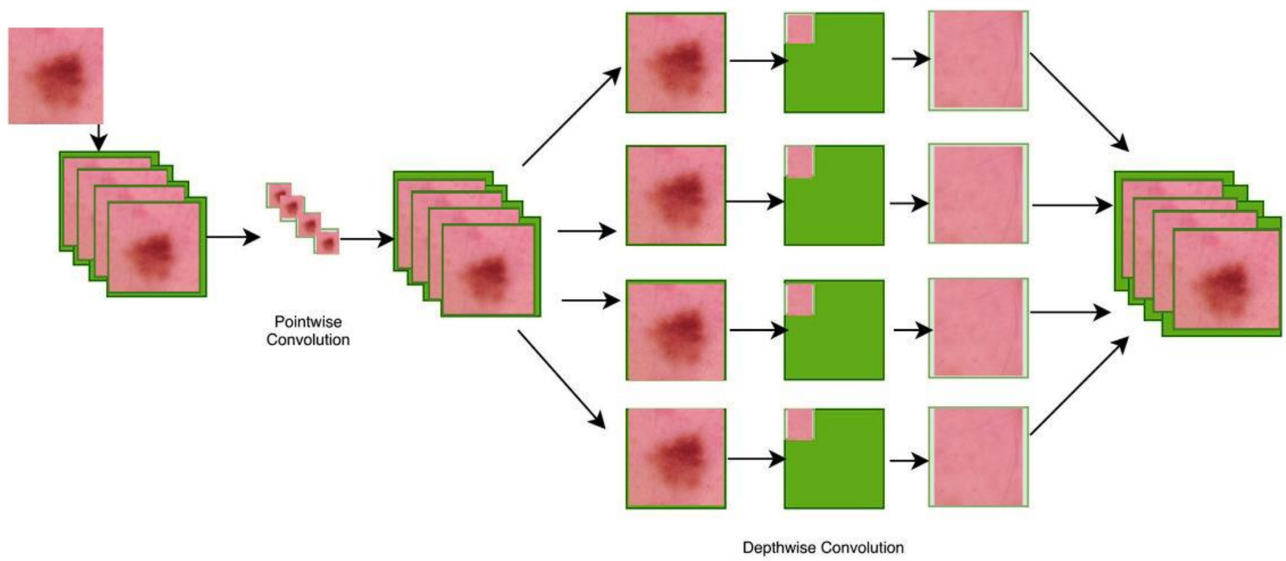

Fig 3.12: Xception working Principle [50]

To meet this challenge, a variety of deep learning networks were experimented upon, including CNN, ResNet, EfficientNet, MobileNet, and Xception. The best performance, 96 percent K validation accuracy, was obtained with Xception [48]. The depthwise separable convolutions of Exception and their ability to represent are one of the features that enable it to outperform the other models in determines their precision and recall when dealing with imbalanced datasets. As the result of these experiments, the model used in the final set of experiments to detect the melanoma is Xception.

Therefore, the model selection procedure reflects a compromise between accuracy, resistance to class imbalance, and computational efficiency, and this way, the HealthQ model will be able to classify the condition reliably in the real-life clinical setting.

3.2.5 System Architecture Choice

Two approaches to the general layout of HealthQ were under consideration: a monolithic one and a modular-service oriented structure.

Monolithic Approach:

In a monolithic system, the application would contain all the features such as appointment booking, queue time prediction, melanoma detection, video consultation, and medicine ordering and the application is tightly coupled. Although this design makes deployment initially easier, it has the disadvantage that scalability, fault isolation, and collaborative development becomes very hard. An example of this is in case one of the modules (e.g., the AI/ML engine) malfunctions, then the whole system becomes inaccessible as well. Moreover, it would be required to redeploy the system to update, or replace one component.

Modular Design (Selected):

In the HealthQ system the approach taken is a more modular, service-oriented one. Every core feature is built as and released as a standalone service:

1. Online Appointment Booking&Queue Management
2. Using ML Engine to predict queue time
3. Melanoma Detection (Automated Intelligence / Deep Learning Engine)
4. Video consultation (API integration with Google Meet)
5. Medicine Ordering

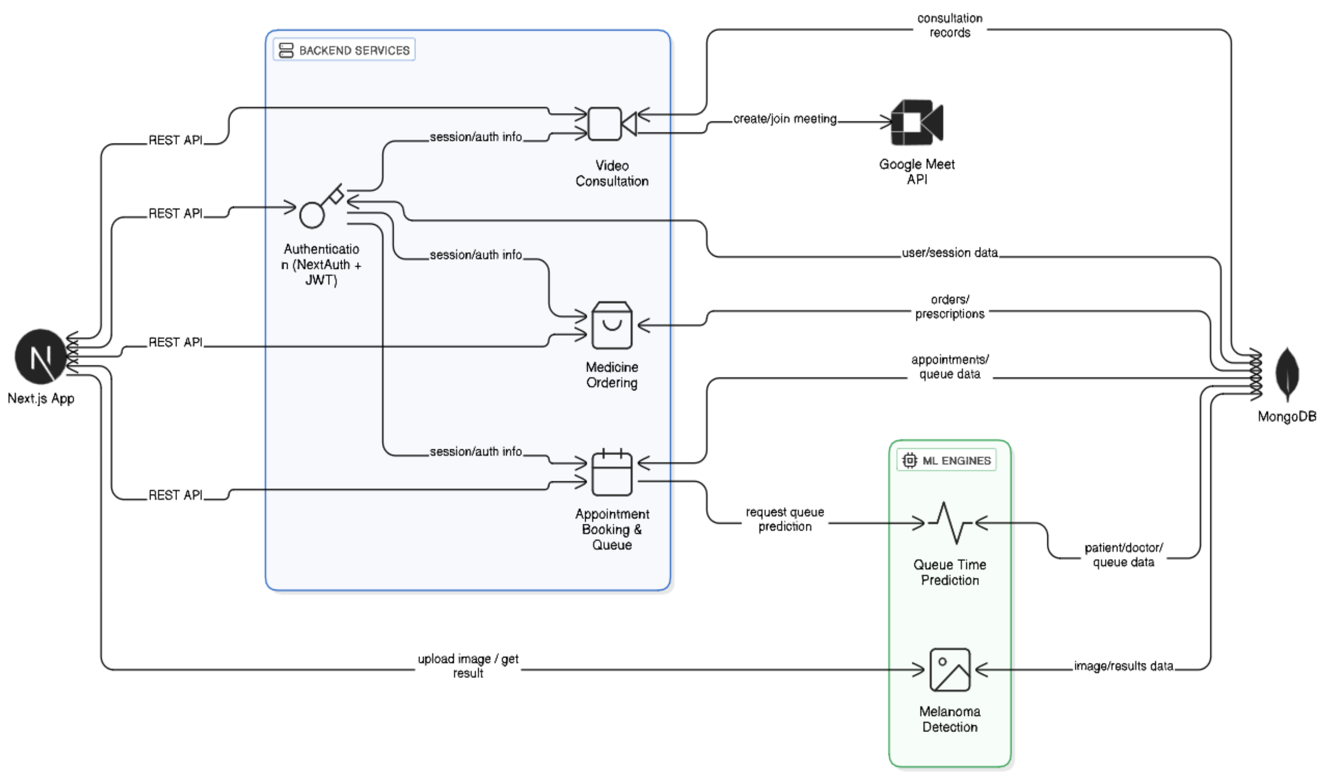

Fig 3.13: Modular Design of HealthQ

These modules would inter-communicate through RESTful APIs, which will enable each of them to be independently developed, deployed, and scaled. As an example, the AI/ML engine can be independently scaled as computational demands increase and without any impact on the appointment booking or video consultation modules.

It is also a strategy that made collaboration in the team highly effective The frontend and backend web development, and AI/ML development were divided to allow working of two people in parallel. This split of the task improved output and decreased development.

The modular approach has been selected since it provides fault tolerance, scalability, parallel development ability, and the long-term maintainability in line with the desire of the HealthQ to become a reliable and extensible healthcare platform.

3.2.6 Deployment & Hosting Strategy

The HealthQ implementation is modular and the application will be a cloud based application therefore it is fault tolerant, flexible as well as efficient in use of resources. Each of the core components is hosted independently thereby reducing the risk of system wide failure and being more maintainable.

1. **Client app (Next.js):** Hosted on Vercel leveraging its Content Delivery Network (CDN), to provide quick query responsiveness and server-side rendering, which is fully integrated by default to optimize search engine optimization and user experience.
2. **Back-end (Express.js APIs):** Running using docker lite containers in Railway/Render. This is the approach that ensures portability, ease and quickness of scaling under loads, and simpler rollbacks in event of failures.
3. **Skin-Cancer Detction**: Xception: Queue prediction + Hosted and deployed with the aid of the FastAPI-based microservices on the cloud machines of EC2/GCP. Where necessary, fine-grained acceleration of graph transformations/node updates is used to achieve high-performance model inference.
4. **MongoDB Atlas**: An API first maintained NoSQL database service that allows flexible schema design, encryption using TLS/SSL, automatic backups and scalability to handle the unpredictable workloads due to use of cloud infrastructure.
5. **Third-Party Services**:Video consultations Google Meet to ensure the quality video communication. Use Google calendar as a convenient way of creating an appointment and getting reminders. Whisper-1 for audio transcription from video consultation.
6. **Pipe Lines/IL/CD Pipe Lines:** The process of combined unit testing, deployment verification and continuous delivery was implemented as a GitHub Action to automate the deployments.
7. **Modular Hosting**: The functionality of each of the modules (appointments, queue prediction, consultations, medicine buying, melanoma detection) is realised separately. This will cause failure of one circuit to be independent of the other.

3.2.7 Challenges & Solutions

In the development of HealthQ, a number of learning curves were experienced in data, model training and hosting, and system integration. The following are the most experienced problems and the solutions to them:

Challenge 1: Imbalanced Dataset

Issue: The HAM10k dataset was highly imbalanced with respect to some classes so it made the predictions biased.

Binarized it into benign and malignant and used an Xception network since it had the highest validation accuracy with imbalanced datasets (96%).

Challenge 2: Forecast of Queue using small real life data

Problem: Reinforcement learning could not be used with the lack of adequate real-time data.

The solution began with a supervised regression model trained by using the sampling data; then the reinforcement learning mechanism was added, which allowed regular updates (daily) using a batch mode that provided continuous fine-tuning without over-saturation.

Challenge 3: Latency of a cloud host

Problem: APIs performed optimally on the local machine, but became slower when deployed on the cloud because they were retrieving additional information that led to slow data-fetching.

Solution: Changed to a modular hosting approach, optimized speed of fetch queries, and made the api response smaller to get faster performance.

Challenge 4: Refraining the third-Party Platform

Problem: It was cost-ineffective to construct own video and scheduling functionality.

Solution: Google Calendar was to be integrated with Google Meet so that they could be used to offer consultation via video as well as booking of an appointment to be reliable and high quality.

Challenge 5: Reliability of a system

Problem: The issue was that when one part collapses the whole platform can collapse.

Solution: Underwent a modular set up where each core element is deployed separately, and the fault isolation is achieved, resulting in increased uptime.

3.3 Project Plan

The project plan consists of the organized process of developing HealthQ, with regard to the phases, timeline and accountabilities that were followed to bring about the completion of this system. It gives an overview of important tasks across the lifecycle of the project start with research, requirements analysis, through the design and development to testing and implementation and makes sure that the project deliverables were completed on time and in a structured way.

3.3.1 Project Phases

The planning of HealthQ development was carried out in multiple stages in order to achieve a systematic flow of project development and efficient use of available resources. These times are

1. Requirement Analysis and Planning -Identifying system requirements, user needs, developing functional requirement and non-functional requirements.
2. System Designing -making architectural diagrams, UI/UX design and determining modular nature of the system.
3. Frontend and Backend Development - Implemetation of the web application using Next.js, connecting the AI/ML API, and establishing database and server-side code.
4. AI/ML Model Development - Curating datasets, pre-processing of the data, training, and selection multiple models, and selecting the model i.e., Xception model based on melanoma detection.
5. Integrating the third-party services, the integration of the Google meet for video consultations and Google calendar to schedule appointments.
6. Testing and Validation-A unit, integration, and user-acceptance testing should be complete to test reliability and accuracy of AI predictions.
7. Deployment and Maintenance - Hosting of the application and monitoring and planning of performance improvements over time.

3.3.2 Timeline

The project plan of HealthQ was well executed with all the stages of the project completed effectively within the stipulated schedule. Instead of time being broken down by days, this scale has the week as the division and weeks overlap where possible to achieve maximum productivity:

Table 3.2: Timeline

Phase	Duration	Key Activities
Requirement Analysis & Planning	Weeks 1–3	Gather requirements, define functional & non-functional specs, initial literature review
System Design	Weeks 4–6	UI/UX design, system architecture, DFD, modular planning
Frontend & Backend Development	Weeks 6–10	Implement Next.js frontend, Express.js backend, database integration

AI/ML Model Development	Weeks 11–14	Data preprocessing, training models (CNN, ResNet, EfficientNet, Xception), selecting best-performing model
Third-Party Integration	Weeks 15–16	Integrate Google Meet & Google Calendar, API connection setup
Testing & Validation	Weeks 16–17	Unit testing, integration testing, AI model validation, user acceptance testing
Deployment & Maintenance	Weeks 17–18	Deploy web app, monitor performance, optimize system, address issues

3.4 Task Allocation

Table 3.3: Task Allocation

Tasks	**Weeks**																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Requirement Analysis																			
System Design																			
Frontend & Backend Development																			
AI/ML Model Development																			
Third-Party Integration																			
Testing & Validation																			
Deployment & Maintenance																			

3.5 Summary

Chapter 3 of the HealthQ system presented how the study will be conducted. It was extensively analyzed by examining system requirements and design specifications, system architecture, functional and non-functional requirements and data flow diagrams. The chapter also addressed the interface design to its patients, clinicians, medicine purchasing and detecting melanoma. The methodology described in detail the choice of the technologies, the development of AI/ML models, and integrations with third parties. The obstacles that were met in the course of development and the way in which they were overcome was explained, the project plan was presented giving phases and schedule. Generally, this chapter gives a detailed guideline on applying and comprehending the HealthQ system.

Chapter 4

Implementation and Results

In this chapter, attention is paid to the practical implementation of the HealthQ system and speaks more about how the developed modules, AI/ML models, and integrations were realized. It brings out development process, implementation of deployment and functioning of each component. Moreover, the results of the testing of the system with regard to predicting the queue time, the accuracy of detecting melanoma, and the general user experience of the web application is provided in the chapter. Such findings prove effectiveness of the approaches and design considerations provided in the prior chapter.

4.1 Environment Setup

HealthQ system was created in accordance with the combination of modern web and AI/ML technologies to be efficient, scalable, and predict the accuracy. The front end was created on Next.js and TypeScript which gave it strong types and the ability to render on the server side, making the app more reliable and an improved UX. The backend webAPI was written in express and is linked to a MongoDB database which is extensible and stores data in documents.

The Python programming language was considered in the development of AI and ML, as it has a wide range of libraries, and simple programming, and the training of various models was carried out in Google Colab Pro with the use of an NVIDIA A100 Graphics Processing Unit due to increased computation speed and deep learning model training. The API keys and sensitive data were stored in variables defined as secure variables, and supports both local development and deployment. This configuration allowed the AI engine and web application to be independent with each other and integrate in an efficient way using APIs.

Table 4.1: Environment Setup for HealthQ Testing

Component	Technology / Tool	Purpose / Details
Frontend	Next.js, TypeScript	Provides server-side rendering, strong typing, and improved user experience.
Backend	Express.js	Handles API routing, server-side logic, and integration with the database.
Database	MongoDB	Stores flexible, document-oriented data for dynamic schemas.
AI/ML Programming	Python	Used for machine learning and deep learning model development.

ML Model Training	Google Colab Pro, NVIDIA A100 GPU	Accelerates computation and allows high-performance model training.
Environment Variables	.env files	Securely store API keys and sensitive information for local and cloud deployment.
Modular Architecture	Yes	Ensures independent functioning of web application and AI engine, connected via API.
Setup for Audio Transcription	Whisper-1 (Axios by OpenAI)	Setting up API keys for cloud access.

4.2 Testing, Evaluation, and Comparative Analysis

4.2.1 Testing & Evaluation

The benefits of the application of the HealthQ system were tried and tested to be reliable, accurate, and performs. Various grades of testing were used:

1. Unit Testing: Every part of the system, comprising of patient management, queue prediction, AI/ML APIs and video consultation, was unit tested to ensure that they are functioning correctly.
2. Integration Testing: The modules were tested together in order to ensure that they allow smooth data flow amongst the frontend, backend and AI/ML engine.
3. Test of the Entire System: the entire system was tested to a simulated real environment such as appointment provision, prediction of the queues, video consultation procedure, and buying of medicines.
4. Usability Acceptance Testing (UAT): The possible users such as patients and physicians provided feedback about their reaction to the dashboards and interactivity based on ease of use, responsiveness and effectiveness.
5. Xception-Based Melanoma Detection Model: The model was tested by using HAM10k data to perform melanoma detection. Accuracy validation was at 96%, where the imbalance present between the malignant and non-malignant lesions were taken into consideration. Models of queue prediction were tested on real world sample data, and the reinforcement learning updates were observed to see a gradual improvement over time in the daily predictions.

Such formal test and evaluation guaranteed that HealthQ satisfies functional requirements, makes accurate predictions and is usable and user-friendly to both patients and healthcare providers.

4.2.2 Performance Analysis

The test of the HealthQ system was performed in several levels, including both models of AI/ML and the application itself. In the melanoma detection module, Xception model produced a validation accuracy of 96%, which is also a good result despite the existence of a class imbalance between malignant and benign samples. The effectiveness of the model was measured using precision, recall, F1-score as well as confusion matrices, whereas the queue prediction model was to be evaluated based on the mean absolute error (MAE), mean squared error (MSE), and R 2 to make sure that wait time forecasts were properly done. The performance of the AI/ML algorithms was quantified by the response time of the API, average latency of queue predictions, and the rate of the consultations of video. The frontend performance, e.g. dashboard loading and interactivity in a scheduled appointment or ordering medicine, was also put to the test. Resource consumption was tracked at the CPU, GPU and memory level in the model predictions and the modular-based architecture of the system provided scalability and reliability with concurrent usage. These performance figures ensure that HealthQ is fast, precise in predictions and the interactions of the user are smooth even in the actual operating environment.

4.3 Results and Discussion

The creation of HealthQ can be used to show how the combination of AI/ML models, modular system design, and real-time web application elements can successfully work to enhance healthcare services. In this section, the findings that are achieved, during the melanoma detection module, queue prediction model and system performance evaluation are discussed. The accuracy of the Xception model is presented in its confusion matrix that marks a high accuracy in the identification of malign and benign cases. The comparative results of other models are summarized in tables of accuracy, and the queue prediction performance in terms of MAE and RMSE. Also, the API response and frontend responsiveness are also measured to assess the overall system performance. Not only do the results provided in this discussion show the raw numbers, but also make them understandable in terms of clinical reality and workflow.

4.3.1 Skin Cancer Detection Accuracy

Table 4.2 Model Accuracy for Skin Cancer Detection

Model Name	Val_Accuracy (%)	Precision (%)	Recall (%)
CNN	82.4	74.4	75.6
InceptionV3	85.32	78.8	78.98
EfficientNetB0	85.09	81.8	80.6
ResNet	90.4	84.45	83.4
Xception	96.6	90	90

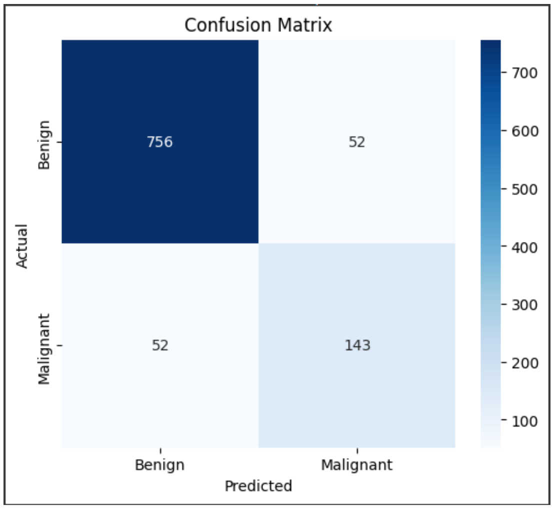

Fig 4.1: Xception Confusion-Matrix

Training Accuracy: 99.98%
Validation Accuracy: 96.6%
Testing Accuracy: 90%

Fig 4.2: Xception Confidence Testing

4.3.2 Queue Prediction API Performance

The queue prediction API used Random Forrest Regressor model to ensure the queue prediction API is accurate, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used in estimating the precision of the ANSYS. The actual time was the same as the predicted result and, therefore, an MAE of 90.5 seconds was calculated which means that the average amount of variance between the actual and the predicted queue times was approximately 1.5 minutes. The RMSE was 118.63 seconds indicating that there were always discrepencies and in some cases were more pronounced. On the whole, the given results indicate that the API is capable of generating reasonably accurate and useful queue time estimates that could be used in the system.

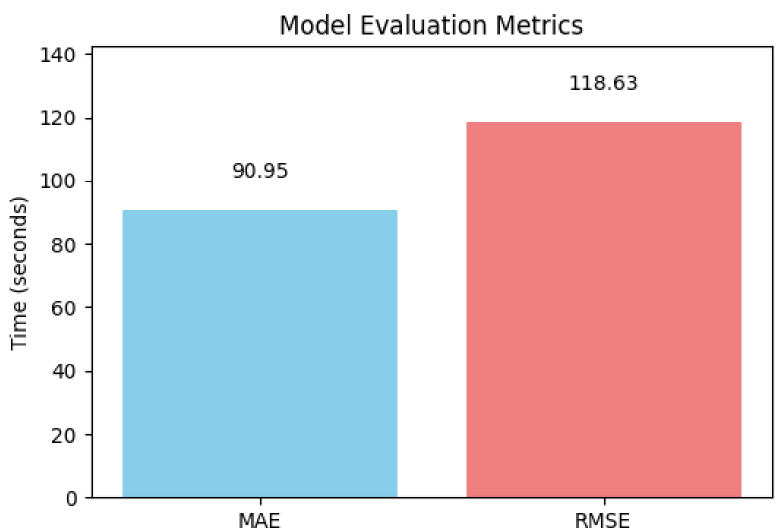

Fig 4.3: MAE and RMSE for Queue Prediction API

4.3.3 Audio Transcription from Video Consultation

Audio Transcription of HealthQ module shows high accuracy and reliability in engaging with video consultations, in English language. That audio recordings done by the doctor-patient sessions are transcribed into text and translated to write in close real time, using the Whisper-1 AI engine, keeping medical terminologies and conversation context. The system has been tested on the standard consultation durations and it shows minimal errors making transcripts usable on patient records and corrective treatment design. As a case in point, a documented consultation fragment which says, Patient complains of frequent heartbeat errors. Probably these are anxiety, stress, consumption of caffeine, and cardiac arrhythmia. Additional scoring with ECG and blood tests prescriptive to eliminate underlying heart conditions. are correctly transcribed and shown in the dashboard, along with vitals and essential information underlined. This performance guarantees that the transcription feature can assist the clinical decision-making process without interfering with privacy and integration with the HealthQ platform.

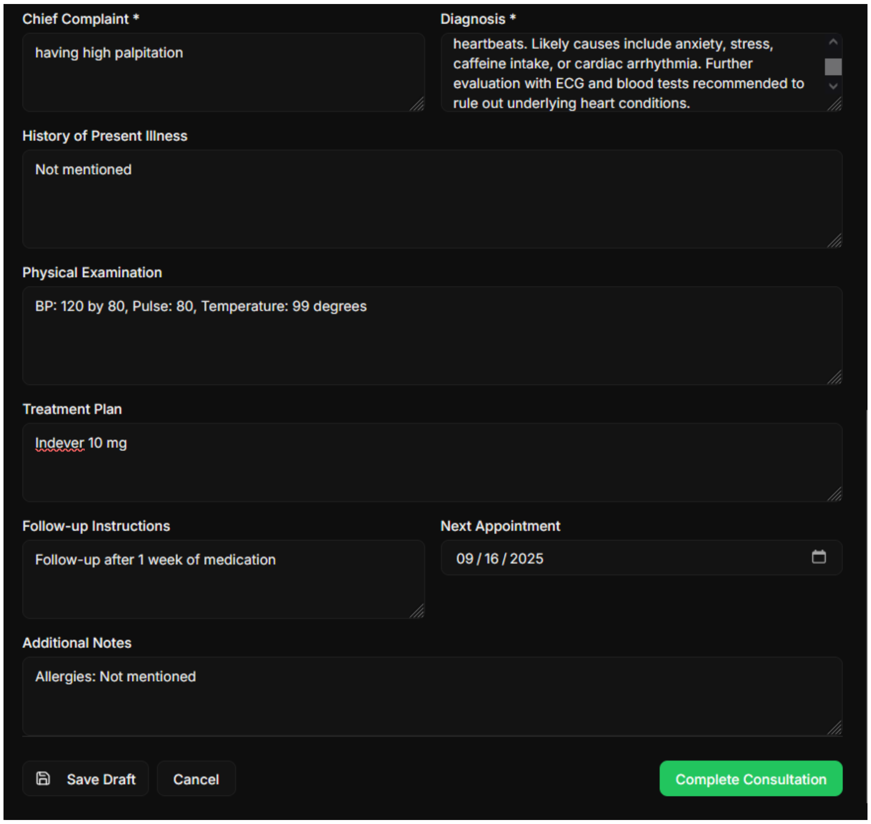

Fig 4.4: Audio Transcription from Video Consultation

4.3 Summary

This division provided the results, and analysis of the implementation of the HealthQ system. The models evaluation revealed that Xception model performed with the highest validation accuracy, though the CNN model, ResNet and EfficientNetB0 performed similarly but a bit lower. The queue prediction API reported a reliable performance with a MAE of 90.5 seconds and an RMSE of 118.63 seconds, and the error is within acceptable limits, which reflects good performance in terms of predicting the patients wait time. It can be concluded that the adopted models and architecture of the system are effective and allow verifying that the proposed solutions to the problems are adequate to their functional requirement and deliver a convenient and convenient experience.

Chapter 5

Engineering Standards & Design Challenges

In this chapter, the standards of engineering adhered to by the engineering, the effects of the HealthQ on society and the environment, and the project management, and financial implications as also the complexities of the engineering problems encountered during development process are highlighted. It brings out the justification of technology-selection, best-practice and ethical implications.

5.1 Compliance with Standards

This section argues about software, hardware, and communication standards to be used in the HealthQ system, as well as alternatives that were taken into consideration and arguments in favor of other decision.

5.1.1 Software Standards

Standards Used:

1. RESTful API Design - Allows the services reached through ML to be modular and scalable.
2. Next.js Best Practices - To ensure a secure and efficient development of full- stack, web applications.
3. MongoDB Data Modeling- To deal with the dynamic healthcare data

Alternates Considered:

Monolithic web apps or modular microservices: microservices have been chosen because this type of application will prevent failure of the entire system and support parallel development.

Rationale:

A modular design is more maintainable and enables the AI/ML modules to be tested in isolation, and is more easily deployable in a scalable manner.

5.1.2 Hardware Standards

Standards Used:

1. Cloud hosted API infrastructure - Avialibilty and scalability. Railway hosting and Hugging Face Space used with octa core processor and 8 GB RAM.
2. Alternatives: NVIDIA A100 GPU-accelerated Servers - NVIDIA A100 GPU to train deep learning models with efficiency.
3. Local GPU vs cloud GPU: Cloud GPU was selected due to effective cost-efficiency and remote access.

5.1.3 Communication Standards

Standards Used:

1. HTTPS/TLS -Keeps patient and doctor information secure in data transmissions.
2. Auth0 2.0 and NextAuth.js -authentication and authorization.
3. Google Meet and Google Calendar Integration -High quality video consultations and booking appointments.

Rationale:

1. Employing standard secure protocols would provide protections required by the healthcare privacy standards and increase trust by the users.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

HealthQ enhances the patient experience by lessening the lines, early detecting melanomas, and conducing hassle-free conferencing. This makes health care better and more accessible to remote locations.

5.2.2 Impact on Society & Environment

By eliminating unnecessarily people visit to the hospitals, HealthQ will contribute to the reduction of resource use and telehealth adoption moreover, reducing transportation-based emissions. Also, increased detection of disease does help in long run for the society.

5.2.3 Ethical Aspects

The security of the patient data and their privacy are taken very seriously because the communication is encrypted and authentication is secure. Explanations of why the I/ML made certain predictions are provided and the system is bias-free through proper handling of datasets (e.g., HAM10k correction).

5.2.4 Sustainability Plan

The modular architecture enables steps of updating and the incorporation of newer AI models. The Cloud infrastructure is scalable in the long run, thus reducing dependency of hardware which is environmental and economically friendly.

5.3 Project Management and Financial Analysis

Project management and financial analysis of HealthQ have adequately balanced susceptibility and capability so that it should have been able to provide sophisticated characteristics at a reasonable cost. The total estimated cost encompassed cloud architecture, the access to the GPUs, the developmental tools, and the mandatory third party additions. The largest amounts were the Google Colab Pro (A100) which supports machine learning model training with Active GPUs, MongoDB Atlas that provides database hosting at scale, Railway through which it can deploy its backends and API, and Hugging Face Hub where it can store and run its models. Google Workspace was integrated to facilitate smooth video consultation and time scheduling that are supplemented by domain and SSL registration to have a secure system entry point. One more important addition was Whisper-1 (OpenAI), which is used to transcribe audio tracks of video grants, which further improves a post-consultation record-keeping system and analytic. This integration provided flexibility to the medical staff as well as made it easier to access exactly by the patients. Unlike TV which required users to incur additional expenses with regards to purchases and rental of personal computers, small operational costs like the internet and subscriptions at minimal cost were also considered. The issue was that the sum of a budget was 200 during four months and it is efficient in the light of the scope and burden of the system. There was also an alternative cost schedule that involved a rapid cut of GPU consumption and cloud reliance, but that approach would restrict the model training time and undermine the responsiveness of real-etime processing such as queue prediction. On the financial side, the product uses a subscription-based business model of clinics and utilises an upgradable nature using premium features like the melanoma detection analytics, queue-time prediction and video transcription. It is affordable and ensures sustainability thereby enabling HealthQ to be used in every small and large healthcare facility.

The table for the budget is given, portraying a comprehensive idea of what services were used and what were the cost of each service accumulated to. The duration also gives us for how long each services has been used.

Table 5.1: Budget Table

Item/Service	Description	Estimated Cost (Monthly)	Duration (Months)	Total Cost (USD)
Google Colab Pro	GPU-based model training (A100 GPU access)	$10	4	$40
MongoDB Atlas (Shared Cluster)	Cloud-hosted NoSQL DB (M0/M10 tiers)	$9	4	$36
Railway Deployment	Hosting for APIs and backend	$5	4	$20
Hugging Face Hub	Model storage & inference API (basic tier)	$9	4	$36
Google Workspace (Meet/Calendar integration)	Business email + meet scheduling	$6	4	$24
Domain & SSL	Custom domain for HealthQ	$12/year (≈ $1/mo)	4	$4
Personal Computer	Used for local dev & testing (no extra cost)	–	–	–
Whisper-1 & OpenAI	Used for audio transcribing	5$ (0.0006$/per min)	4	$20
Miscellaneous	Internet, minor subscriptions/tools	$5	4	$20
Total (Actual Budget)				**$200**

5.4 Complex Engineering Problems

This section will dig into the project in terms of complex engineering problems and activities. The practice of HealthQ Development of Heal Fast, Queue Less will not only use machine learning algorithms, but also with cloud services, database management, and health care workflows. To this end, this project should be mapped to the problem categories in engineering, knowledge profiles and engineering activities to show the complexity and the depth of knowledge it encompasses.

5.4.1 Complex Problem Solving

HealthQ system solves a complex engineering challenge where AI-based diagnosis, real-time queue forecasting and hallmark telehealth tools were merged into a single unit. It is also complex to balance technical accuracy, usability and healthcare standards and necessitate systematic solutions to problems spanning a variety of engineering disciplines.

Table 5.2: Problem Solving Table

EP1 Depth of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✓	✓	✓	✓		✓	

EP1: HealthQ project demands expertise at the specialist level due to the implementation of methologies of machine learning and deep learning, namely the use of state-of-the-art models, i.e., CNN, Xception, InceptionV3, ResNet, and EfficientNet. These models were tested and contrasted to achieve good quality of detection of melanoma with good accuracy. Besides model training, the project required the API development and continuity integration with databases like MongoDB Atlas and with the cloud deployment softwares like Railway and Hugging Face. The system also uses regression model-related prediction of queue, thus needing expertise on a measure of evaluation as the MAE and the RMSE. On the frontend, the patient and doctor dashboards built using React.js, demonstrate higher order of UI/UX development capabilities. Also, the system combines Whisper-1 to be able to transcribe recorded consultations by speech-to-text and, therefore, had to work with audio processing, secure API calls, and natural language conversion. The project provides a level of knowledge who combined computer vision, NLP, cloud computing, and frontend engineering well exceeds undergraduate fundamentals, and points to the variety of knowledge required to implement the systems tested in the real world of healthcare.

EP2: The implementation and design of the project had to meet a number of opposing requirements. As an example, scaled sinuosity models like Xception needed to be trained with greater computational intensity to attain extremely high melanoma detection performance, potentially raising the cost of inference and system overhead. Conversely, the timeliness matter in real-time performance is essential to the prediction of queues and teleconsultation, which implies the balance between efficiency and accuracy. Equally, good data protection input and standard requirements such as HIPAA/GDPR are in conflict with quick and slim deployments. One more clash was the monetary one, which was to make decisions between cheap options as Google Colab Pro to train and expensive ones as dedicated GPUs to get better exposure. These trade-offs had to be optimized properly in such way, that they were to have a balance in terms of speed, accuracy, usability, and cost-efficiency. Finally, HealthQ shall overcome these contradictions by choosing the best models in each use case at a time leaving the deployment reasonably priced and scaleable.

EP3: The proposal required it to conduct a large amount of comparative studies in the influx of deep learning models (Xception, InceptionV3, ResNet, EfficientNet, CNN) that detected melanoma. All models were assessed in terms of accuracy, precision, recall and F1-score where queue model prediction was evaluated using MAE and RMSE. In addition to classification and prediction the integrations of Whisper-1 and audio transcription gave a second layer of analysis, entailing the comparison of the accuracy of transcription, latency and coping of medical terms during a consultation. This ensured that extracted text can be trusted and that is contextually current so that it can be utilized in a healthcare documentation. The multi-layered evaluation of the image analysis, queue prediction, and speech-to-text typing are the level of analysis that is far serious than the approaches to problem being solved manually and involves a complex degree of expertise about the different components of AI, along with its interrelation in the environment that is sensitive to health care.

EP4: The problems that are considered by the HealthQ system are comparatively novel and unknown both technically and medically. Implementation of AI in clinical work process is very new and begs the question of privacy of patient data, ethical utilization of AI and interpretability of the system. Not all these problems are amenable to known solutions, like how to handle black box deep learning models or how to make physicians trust AI systems. The telehealth feature integration with automated diagnosis also presents some additional complexities with video data storage, privacy of transcription, and real time functionality. These problems are not entirely recorded in known ideas of engineering practices thus to resolve them aide will be necessary in the form of adaptability, creativity and sensitivity to ethics. This is a lack of familiarity which elaborates the complexity in the project and makes HealthQ a futuristically oriented healthcare solution.

EP6: The HealthQ project cannot be successful and usable without the involvement of several stakeholders. The main stakeholders are patients because their interactions with the system like appointments, queue positions, and access to video consultation transcripts specifically dictate system efficiency. Interpretation of transcripts of consultations and scheduling of appointments are also extremely important functions of doctors and healthcare providers to verify the results of melanoma detection. Technical stakeholders, which consist of developers and machine learning engineers, will work to provide the robustness, accuracy, and functionality of the platform. Alongside, third-party service providers like Google Meet, Google Calendar, or Whisper-1 to identify transcriptions are indirect stakeholders, as their functionality affects the functionality of the system. Lastly, regulatory bodies and health facilities are not directly connected but still important stakeholders where their policies touch on compliance and ethical factors. This variety of stakeholders highlights the multifaceted nature of the project, which necessitates interaction and additional feedback and change design to consider the demands of each stakeholder.

To conclude, the HealthQ project qualifies as a good solution in several issues of complex problem solving listed in the engineering standards. The mix of technical and societal dimensions of depth in the project is evidenced by using expertise in deep learning (Melanoma detection in Xception), practical database and system integration experience (MongoDB, APIs, React.js), and addressing ethical treatment of healthcare data. The off-tradeoffs of accuracy, speed, cost are a realistic engineering problem, whereas adherence to standards like HIPAA/GDPR demonstrates the understanding of international practices. Lastly, the inclusion of new technologies such as Whisper-1 transcription emphasizes the concept of innovation and flexibility when it comes to tackling healthcare-related issues.

5.4.2 Knowledge Profiles

Table 5.3: Knowledge Profiles Mapping

K1 Natural Science	K2 Mathematics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K7 Comprehension	K8 Research Literature
	✓	✓	✓	✓	✓	✓	✓

K2: The project utilized statistics and probability in order to assess the distribution of the datasets and take care of the imbalance of classes (benign vs malignant). Queue prediction was evaluated using metrics of MAE and RMSE whereas Precision, Recall and F1-score were used to evaluate the performance of the model. CNN/Xception backpropagation comprehension relied on the illumination of linear algebra and calculus.

K3: There was a necessity to have a solid understanding of algorithms and database foundations. MongoDB Atlas was essential in providing necessary efficiency in retrieving and storing data, whereas query optimization provided scalability. Both the deep learning pipelines and Whisper-1 transcription handling relied on using algorithmic knowledge. This groundwork allowed the seamless system integration of models alongside API.

K4: Specialized expertise in deep learning models (CNN, Xception, InceptionV3, EfficientNet, ResNet) was applied for melanoma detection. Whisper-1 required knowledge of transformer-based speech-to-text systems. Additionally, React.js was used for frontend development, while backend services integrated with ML APIs and MongoDB databases. This combination of advanced ML and software engineering tools highlights the project's reliance on specialist knowledge.

K5: The project involved designing an integrated system architecture combining API + DB + models. Frontend development with React.js provided interactive dashboards for patients and doctors. MongoDB Atlas was chosen for its flexibility with healthcare data storage. The design also included the video consultation pipeline recording, extracting audio, transcribing via Whisper-1, and storing results ensuring seamless workflow across modules.

K6: To be implemented meant practice of theory via Google Colab Pro (model-training), Railway deployment (API hosting), and Hugging Face Hub (model storage/tracking). The Whisper-1 transcription has been tested on real world consultant recordings to confirm accuracy. Production dashboards constructed with React.js + MongoDB served as useful real-world engineering experience.

K7: Within the project, understanding the aspects of the problems related to society and ethical issues in digital healthcare needed to be conducted, with an increased focus on the privacy of patient data, the safety of exploring videos and confidential transcription. The introduction of Whisper-1 raised some apprehensions in regard to delicate audio information administration. Next to technical work, understanding also went to interpreting medical ML research and trade-offs with performance to benefits of the clinics.

K8: A thorough review of melanoma detection research guided model choice (Xception for best accuracy with imbalanced HAM10k data). Studies on queue prediction methods informed the regression model evaluation. Research on speech-to-text transcription validated the choice of Whisper-1 for medical consultation transcriptions. Literature on React.js + MongoDB in healthcare systems supported design decisions for a scalable and user-friendly platform.

5.4.3 Engineering Activities

Table 5.4: Engineering Activities

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓	✓	✓	✓	

EA1: HealthQ application uses numerous computational and software environments, such as Google Colab Pro A100 with a needful GPU to train the model, MongoDB Atlas to handle healthcare information, Railway and Hugging Face to deploy APIs and ML models, and React.js as a frontend developer. Besides, the Whisper-1 is incorporated with transcription services. It is this rich array of distributed resources that allow both real-time and offline capabilities to be provided effectively.

EA2: The project will involve multi-hierarchy between developers, medical workers, and patients. Developers do system coding, deployment, and system design, doctors verify if detecting melanoma model works and is accurate, patients use dashboards in order to book appointments, track queues and also to have a consultation date. This degree of interaction will make the design user-centered with different stakeholders providing feedback to influence the system.

EA3: HealthQ is a innovative project applying AI-based melanoma detection with queue management and video consultation transcription into one unified class of healthcare architecture. In contrast to conventional healthcare applications, which concentrate on telehealth or appointments, HealthQ is the only one to integrate predictive modeling, machine learning-based diagnostics, and online healthcare processes. This is a hybrid solution that will not only make clinicsefficient, and provide high-level diagnostic assistance to patients.

EA4: The system adds value to the society by decongesting hospitals as well as bringing efficiency in the queues and also early detection of melanoma that can save lives. Furthermore, it reduces the environmental impact because it digitizes the healthcare processes, eliminating unreasonable movement and waiting time. The project thereby establishes a two-fold good, which is societal enhancement of healthcare delivery and sustainability due to smarter use of resources.

5.5 Summary

In this chapter, the author discussed the engineering standard, social impacts, economics, and cost-complexity analysis of HealthQ - Heal Fast, Queue Less project. We addressed compatibility and connection to standard computer software, hardware and communications protocols, with justification of the choice of technologies. The impact of the project on healthcare, society, ethics, and sustainability was examined to show its greater scope. A cost analysis and an alternative budget was provided to analyze the financial viability of the solution. In addition, the project was cross referenced to complex engineering problem characteristics, knowledge base, and engineering activities defining the project as the complex engineering project. Of all the advantages of the system, this chapter sustains the significance of the system and the technical rigor, which are of major significance in terms of system implementation and sustainability.

Chapter 6

Conclusion

This chapter provides a description of the overall accomplishments of the HealthQ, Heal Fast, Queue Less, project, identifies the limitations of the project, and gives a possible future work. It comments on the products of our system, both on the technical side of things, and on their usability as a practical matter

6.1 Summary

The project was able to come up with a holistic healthcare system with the functionality of making appointments, queuing, and applying melanoma detection via the use of AI. Through the use of the latest technology in machine learning the proposed solution demonstrated high performance in melanoma classification based on the latest machine learning techniques like Xception (trained on HAM10k dataset). In addition, the design and testing of the queue prediction API were based on data containing elements of real world situations with the results not exceeding acceptable limits in terms of MAE and RMSE. The implementation will lead to the benefit of the lessened waiting time and better decision-making on the sides of patients and healthcare providers.

6.2 Limitation

Despite its success, the project has a few limitations:

1. Dataset limitations: The HAM10k dataset, while widely used, is limited in size and scope, and class imbalance remains an issue.
2. Queue prediction constraints: The model predictions depend heavily on simulated data and may require real-world hospital data for better accuracy.
3. Scalability: While the system is functional, large-scale deployment in hospitals with thousands of patients may require optimization and additional cloud resources.
4. Integration challenges: Real-time integration with hospital systems (e.g., EMRs, patient databases) has not been fully tested.

6.3 Future Work

In future development, the following direction can be given.

1. Larger and more variety of datasets: Using real-world clinical data that would cover melanoma and hospital queues time further enhancing predictions.
2. State-of-the-art ML: This topic includes the investigation of the SMOTE algorithm, GAN-based augmentation, or focal loss to deal with class imbalance, which hinders melanoma detection.
3. Mobile application:Making it available as a mobile application, easier to use by the patient and notify them.
4. Connectivity to IoT/Smart hospital systems: Interconnecting with wearables, or monitoring devices in the hospital.
5. Scalability testing - Improved performance to work with higher number of patients and hospitals at a time.

References

[1] Digital Adoption, Digital Healthcare Research Report, 2024. [Online]. Available: https://www.digital-adoption.com/digital-adoption-healthcare-research/

[2] L. Pumplun, et al., "*Adoption of machine learning systems for medical diagnostics in clinics,*" Journal of Medical Internet Research, vol. 23, no. 3, p. e27107, 2021. doi: 10.2196/27107.

[3] M. Hassan, et al., "*Barriers to and facilitators of artificial intelligence adoption in health care: A scoping review*," JMIR AI, vol. 3, no. 1, p. e47629, 2024. doi: 10.2196/47629.

[4] E. D. Bărcanescu, "*Artificial intelligence in digital healthcare systems*," Applied Sciences, vol. 11, no. 13, p. 5942, 2021.

[5] S. P. Bhavnani, et al., "*Digital health: A path to validation*," npj Digital Medicine, vol. 3, no. 1, p. 2, 2020.

[6] S. M. S. Islam, et al., "*eHealth in Bangladesh: Current status, challenges, and future direction*," Frontiers in Public Health, vol. 8, p. 460, 2020.

[7] T. Cayirli and E. Veral, "*Outpatient scheduling in health care: A review of literature*," Production and Operations Management, vol. 12, no. 4, pp. 519–549, 2003.

[8] D. Gupta and B. Denton, "*Appointment scheduling in health care: Challenges and opportunities*," IIE Transactions, vol. 40, no. 9, pp. 800–819, 2008.

[9] J. Patrick, M. L. Puterman, and M. Queyranne, "*Dynamic multi-priority patient scheduling for a diagnostic resource*," Operations Research, vol. 56, no. 6, pp. 1507–1525, 2008.

[10] A. Naeem, M. S. Farooq, A. Khelifi, and A. Abid, "*Malignant Melanoma Classification Using Deep Learning: Datasets, Performance Measurements, Challenges, and Opportunities*," IEEE Access, vol. 8, pp. 110575–110597, 2020.

[11] R. Indraswari, R. Rokhana, and W. Herulambang, "*Melanoma Image Classification Based on MobileNetV2 Network*," Procedia Computer Science, vol. 197, pp. 198–207, 2022.

[12] A. N. Richter and T. M. Khoshgoftaar, "*Modernizing Analytics for Melanoma with a Large-Scale Research Dataset*," in IEEE IRI, pp. 551–558, 2017.

[13] S. Mukherjee, A. Adhikari, and M. Roy, "*Malignant Melanoma Classification Using Cross-Platform Dataset with Deep Learning CNN Architecture*," Springer Singapore, pp. 31–41, 2019.

[14] V. Rotemberg, et al., "A Patient-Centric Dataset of Images and Metadata for Identifying Melanomas Using Clinical Context," Scientific Data, vol. 8, no. 1, p. 34, 2021.

[15] N. Auslander, et al., "*Robust Prediction of Response to Immune Checkpoint Blockade Therapy in Metastatic Melanoma*," Nature Medicine, vol. 24, no. 10, pp. 1545–1549, 2018.

[16] S. M. Thomas, et al., "*Non-Melanoma Skin Cancer Segmentation for Histopathology Dataset*," Data in Brief, vol. 39, p. 107587, 2021.

[17] S. J. Soong, et al., "*Predicting Survival Outcome of Localized Melanoma: An Electronic Prediction Tool Based on the AJCC Melanoma Database*," Annals of Surgical Oncology, vol. 17, pp. 2006–2014, 2010.

[18] R. A. Scolyer, et al., "*Data Set for Pathology Reporting of Cutaneous Invasive Melanoma: Recommendations from the International Collaboration on Cancer Reporting (ICCR)*," The American Journal of Surgical Pathology, vol. 37, no. 12, pp. 1797–1814, 2013.

[19] S. A. Leachman, et al., "*Methods of Melanoma Detection*," Melanoma, pp. 51–105, 2016.

[20] H. Tsao, et al., "*Early Detection of Melanoma: Reviewing the ABCDEs*," Journal of the American Academy of Dermatology, vol. 72, no. 4, pp. 717–723, 2015.

[21] A. A. Adegun and S. Viriri, "*Deep Learning-Based System for Automatic Melanoma Detection*," IEEE Access, vol. 8, pp. 7160–7172, 2019.

[22] G. Argenziano, et al., "*Accuracy in Melanoma Detection: A 10-Year Multicenter Survey*," Journal of the American Academy of Dermatology, vol. 67, no. 1, pp. 54–59, 2012.

[23] S. M. A. Weinstock, "*Early Detection of Melanoma*," Jama, vol. 284, no. 7, pp. 886–889, 2000.

[24] A. G. Goodson and D. Grossman, "*Strategies for Early Melanoma Detection: Approaches to the Patient with Nevi*," Journal of the American Academy of Dermatology, vol. 60, no. 5, pp. 719–735, 2009.

[25] E. Okur and M. Turkan, *"A Survey on Automated Melanoma Detection*," Engineering Applications of Artificial Intelligence, vol. 73, pp. 50–67, 2018.

[26] S. H. Kassani and P. H. Kassani, "*A Comparative Study of Deep Learning Architectures on Melanoma Detection*," 2019.

[27] E. Vocaturo, D. Perna, and E. Zumpano, "*Machine learning techniques for automated melanoma detection*," in 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Nov. 2019, pp. 2310–2317.

[28] R. Patil and S. Bellary, "*Machine learning approach in melanoma cancer stage detection*," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 6, pp. 3285–3293, 2022.

[29] H. Bhatt, V. Shah, K. Shah, R. Shah, and M. Shah, "*State-of-the-art machine learning techniques for melanoma skin cancer detection and classification: A comprehensive review*," Intelligent Medicine, vol. 3, no. 3, pp. 180–190, 2023.

[30] C. González Cruz, M. A. Jofre, S. Podlipnik, M. Combalia, D. Gareau, M. Gamboa, M. G. Vallone, Z. F. Barragán-Estudillo, A. L. Tamez-Peña, J. Montoya, and M. A. Jesús-Silva, "*Machine learning in melanoma diagnosis. Limitations about to be overcome*," Actas Dermo-Sifiliográficas (English Edition), vol. 111, no. 4, pp. 313–316, 2020.

[31] T. Safran, A. Viezel-Mathieu, J. Corban, A. Kanevsky, S. Thibaudeau, and J. Kanevsky, "*Machine learning and melanoma: the future of screening*," Journal of the American Academy of Dermatology, vol. 78, no. 3, pp. 620–621, 2018.

[32] M. M. Rahman, M. K. Nasir, A. Nur, S. I. Khan, S. Band, I. Dehzangi, A. Beheshti, and H. A. Rokny, "*Hybrid feature fusion and machine learning approaches for melanoma skin cancer detection*," 2022.

[33] L. Li, Q. Zhang, Y. Ding, H. Jiang, B. H. Thiers, and J. Z. Wang, "*Automatic diagnosis of melanoma using machine learning methods on a spectroscopic system*," BMC Medical Imaging, vol. 14, pp. 1–12, 2014.

[34] S. Mustafa, A. B. Dauda, and M. Dauda, "*Image processing and SVM classification for melanoma detection*," in 2017 International Conference on Computing Networking and Informatics (ICCNI), Oct. 2017, pp. 1–5.

[35] H. Alquran, I. A. Qasmieh, A. M. Alqudah, S. Alhammouri, E. Alawneh, A. Abughazaleh, and F. Hasayen, "*The melanoma skin cancer detection and classification using support vector machine*," in 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Oct. 2017, pp. 1–5.

[36] M. I. A. Al-Mashhadani, O. N. Ucan, and O. Bayat, "*Melanoma cancer analysis towards early detection using machine learning algorithms*," Journal of Medical Imaging and Health Informatics, vol. 9, no. 3, pp. 514–527, 2019.

[37] X. Dai, I. Spasić, B. Meyer, S. Chapman, and F. Andres, "*Machine learning on mobile: An on-device inference app for skin cancer detection*," in 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Jun. 2019, pp. 301–305.

[38] A. Hussaindeen, S. Iqbal, and T. D. Ambegoda, "*Multi-label prototype based interpretable machine learning for melanoma detection*," International Journal of Advances in Signal and Image Sciences, vol. 8, no. 1, pp. 40–53, 2022.

[39] O. T. Jones, R. N. Matin, M. Van der Schaar, K. P. Bhayankaram, C. K. I. Ranmuthu, M. S. Islam, D. Behiyat, R. Boscott, N. Calanzani, J. Emery, and H. C. Williams, "*Artificial intelligence*

and machine learning algorithms for early detection of skin cancer in community and primary care settings: a systematic review," The Lancet Digital Health, vol. 4, no. 6, pp. e466–e476, 2022.

[40] DocTime, "DocTime – Online Doctor Appointment Booking," DocTime, 2025. [Online]. Available: https://www.doctime.com

[41] Popular Diagnostic, "Popular Diagnostic – Book Lab Tests & Appointments," Popular Diagnostic, 2025. [Online]. Available: https://www.populardiagnostic.com

[42] Ibne Sina Healthcare, "Ibne Sina Healthcare – Online Appointments & Services," Ibne Sina Healthcare, 2025. [Online]. Available: https://www.ibnesinahealthcare.com

[43] Practo, "Practo – Book Doctor Appointments & Consult Online," Practo, 2025. [Online]. Available: https://www.practo.com

[44] Doctolib, "Doctolib – Online Doctor Appointments & Teleconsultations," Doctolib, 2025. [Online]. Available: https://www.doctolib.com

[45] Maya, "Maya – Health Assistant & Online Consultations," Maya, 2025. [Online]. Available: https://www.maya.com

[46] OpenAI, "Whisper: Automatic Speech Recognition System," OpenAI, 2025. [Online]. Available: https://openai.com/research/whisper

[47] Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data, 5, 180161. https://doi.org/10.1038/sdata.2018.161

[48] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251–1258. https://doi.org/10.1109/CVPR.2017.195

[49] Eraser.io, "Collaborative online diagramming tool," [Online]. Available: https://eraser.io

[50] Canva, "Visual communication platform," [Online]. Available: https://www.canva.com

HealthQ - AI Powered Smart Healthcare Platform for Appointments & Disease Detection

ORIGINALITY REPORT

2%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%
2	Submitted to Daffodil International University Student Paper	<1%
3	www.coursehero.com Internet Source	<1%
4	link.springer.com Internet Source	<1%
5	www.frontiersin.org Internet Source	<1%
6	www.mdpi.com Internet Source	<1%
7	Submitted to United International University Student Paper	<1%
8	Fayadh Alenezi, Ammar Armghan, Kemal Polat. "A multi-stage melanoma recognition framework with deep residual neural network and hyperparameter optimization-based decision support in dermoscopy images", Expert Systems with Applications, 2023 Publication	<1%

9	Wasswa Shafik, Adel Ben Youssef, Chithirai Pon Selvan, Pushan Kumar Dutta. "Sustainable Healthcare Systems in Africa - Technologies, Practices, and Management", Routledge, 2025 Publication	<1 %
10	discovery.researcher.life Internet Source	<1 %
11	laganvalleydup.co.uk Internet Source	<1 %
12	dalspace.library.dal.ca Internet Source	<1 %
13	jmhg.springeropen.com Internet Source	<1 %
14	www.biorxiv.org Internet Source	<1 %
15	www.tandfonline.com Internet Source	<1 %
16	dspace.mit.edu Internet Source	<1 %
17	dokumen.pub Internet Source	<1 %
18	formations.univ-smb.fr Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.