Chicken disease detection web app using AI

BY

Name:Hasibul Islam

ID: 201-16-528

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computing And Information System.

Supervised By

Md. Nasimul Kader

Assistant Professor

Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

23, SEPTEMBER 2024

APPROVAL

This thesis titled **"Chicken Disease Detection Web App using AI "**, Submitted by **Md. Hasibul Islam**, ID No: **201-16-528**, to the Department of Computing & Information System, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information System and approved as to its style and contents. The presentation has been held on 23-09-2024.

BOARD OF EXAMINERS

Md Sarwar Hossain Mollah — **Chairman**
Associate Professor and Head
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Md. Nasimul Kader — **Internal Examiner**
Assistant Professor
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Md. Mehedi Hassan — **Internal Examiner**
Lecturer
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Dr. Saifuddin Md. Tareeq — **External Examiner**
Professor
Department of Computer Science and Engineering
University of Dhaka, Dhaka

DECLARATION

I, **Md. Hasibul Islam** hereby declare that; the work in this dissertation titled **"Chicken Disease Detection Web App Using AI"** has been done by me under supervision of Mr. **Md. Nasimul Kader**, Assistant Professor, department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By

Mr. Md. Nasimul Kader
Assistant Professor
Department of CIS
Daffodil International University

Submitted By

Md. Hasibul Islam
ID: 201-16-528
Department of CIS
Daffodil International University

ACKNOWLEDGEMENT

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

We are grateful and wish our profound indebtedness to**, Associate Professor, Department of CIS Daffodil International University, Dhaka.** The Deep knowledge & keen interest of my supervisor in the field of "Web development" helped to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express my heartiest gratitude to **Head,** Department of CIS, for his kind help to finish my project and also to other faculty members and the staff of the CIS department of Daffodil International University.

We would like to thank our entire course mate 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

You can read the entire findings of my proposal, a "Chicken disease detection web app using AI," here. This article goes into detail on the processes used to develop the idea into a working website. The user dashboard represents a particular element that sticks out to system users. Chicken diseases are very harmful for every chicken farm for their business. Because chicken producers have little opportunity for agricultural support services, these illnesses do not show symptoms in their flocks of birds early. Deep learning methods may be able to identify certain poultry illnesses early on. This website will require user input in order to identify chicken diseases. Thus, there will be two basic components to it. One involves developing websites, while the other involves using deep learning to identify chicken diseases. In this work, there have been used 4 classes of chicken diseases for detection using web application like: “Salmonella”, “Coccidiosis”, “New Castle disease” and “Healthy”. To predict and detect chicken sickness in order to categorize utilizing chicken coop diseases, four models are used: CNN, VGG16, MobileNetV3, and Resnet. Resnet achieves the greatest accuracy of 81.70%. Ultimately, the Resnet network is used for classification to identify chicken sickness and provide a web app. The report covers all aspect of the web application's creation procedure, from conception to execution, including its structure, user interface fashion, and the technologies used. We used Django with Python for the backend, JavaScript was used for the user interface. Our system application may be set up with just a standard desktop machine and internet access; costly software or computer components are not required.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Introduction

Apps that show the relationships and interactions amongst different apps are called systems. Programming connections, applications, and system management tools are all found in the "System" page of computers. While the term "system" may have varied meanings depending on the situation, the idea is basically the same. The "Chicken disease detection web app using AI" builds a thorough foundation by integrating a number of different technologies. Each system under examination has its limits applied by the several modules that comprise this framework. Numerous systems are present in every module. It provides a wide range of services, such as a log-in system and a detection system, through this online application.

The production of poultry contributes significantly to the expansion of the world economy and is essential to the sustainability of the human ecology. It provides inexpensive and easily obtainable forms of protein, including meat, eggs, and various other derivatives. Poultry production contributes to global economic progress not just via direct nutritional advantages but also by increasing household income, promoting food security, and assisting in the eradication of poverty. However, the demand for chicken meat and eggs rises along with the world's population. At the same time, managing poultry diseases becomes a major task that poses serious risks to both economic stability and food security. Using state-of-the-art technology provides opportunities to develop approaches that improve farm profitability while simultaneously reducing environmental effects and promoting livestock and human welfare.

This article explains the clinical symptoms associated with different diseases by doing a thorough analysis of the most recent research on deep learning-based poultry illness detection. According to the report, when it comes to early illness identification and warning in the poultry industry, emerging technology solutions—particularly the use of image processing and deep learning (DL)—perform noticeably better than traditional human inspection methods also develop web app for predict chicken diseases. These developments highlight their potential to completely transform the mitigation of illness and control of

poultry health. Our website uses authentication techniques so that only legitimate users may access specific sections of it. It provides a wide range of services, including a detection system, through the use of this web-based application.

CHAPTER 2

Initial Study

2.1 Project Proposal

Objectives

This project's primary goal is to develop an intuitive online tool that can correctly recognize and diagnose illnesses in hens using pictures or certain symptoms. Our goal is to give farmers a dependable and effective tool that may help with early illness identification, lower the risk of financial losses, and enhance animal wellbeing by utilizing AI approaches.

- To use deep learning to predict the five different types of chicken illnesses using coop pictures.
- To compile information for the purpose of predicting illnesses in chickens.
- Acquiring a comprehensive understanding of the domains related to deep learning.
- Applying a range of strategies to enhance outcomes.

Benefits of the website:

- This project has been used for prediction chicken disease using chicken coop.

2.2 Background of the Project

The built-in deep learning model can help with the evaluation and oversight of the poultry farming business, particularly in the early diagnosis of sick poultry, thanks to the capacity of deep learning technology to self-learn how to evaluate photos. Deep learning and driven artificial intelligence (AI) techniques might be integrated into the statistical analysis procedure to identify, forecast, and notify end users of unusual situations in order to prevent the spread of poultry illnesses and maintain biosecurity. The utilization of big data offers an unparalleled prospect for the creation of instruments that will maximize agricultural income, diminish ecological footprints, and improve the well-being of both people and animals [9].Identifying anomalies in poultry and providing

notification in advance of infectious illnesses are critical for maintaining animal health and minimizing losses.

Rich details regarding a poultry's mental wellness, physical state, and surroundings are included in its physiological characteristics. Use of this data can help with decision-making about husbandry techniques and chicken welfare monitoring. Physiological features have the potential to be employed as a means of monitoring environmental and animal health changes, as demonstrated by studies. Numerous illnesses are linked to basic physiological characteristics in poultry, including as temperature of the body, speaking, and feces [10]. To reduce risks, these variables are monitored and used to identify and diagnose illnesses early on.

2.3 Problem Area

When using JavaScript to build a website for the identification of chicken diseases, there are quite a few potential problems to consider. By addressing these potential issues early on in the process of constructing a detecting website using Django with Python, you may improve user experience and reduce the likelihood of serious issues after launch. It seems that combining and assessing all of those data sets is the main goal of this research. With a range of tools and methods, the collection of data had been cleaned up and standardized. The sheer size of the data sets and the fact that they included several layers from various historical periods meant that we had to wait a long time to see the final product. There are several sources of datasets on this subject. Because we didn't finish the research, I find it challenging to quickly select the best responses because each assignment requires a lot of hard work on my side. Another problem was preserving this enormous amount of data and getting it ready for classification. The task involved classifying the component elements of chicken illnesses in an online application using deep learning models.

2.4 Possible Solution

- Design adaptable to many screen sizes.

- User-friendly and straightforward website features, accompanied by a decent user interface.
- I will illustrate how the study on chicken illness coops serves as the foundation for various chicken diseases.
- A deeper comprehension of how to identify disease cartilage using DL from images of ill coops.
- Using old picture data, one of my objectives is to contrast my results with earlier studies.
- Yet another is to determine which CNN model with DL, depending on data availability, is most effective in identifying various chicken illnesses.

CHAPTER 3

Literature Review

3.1 Discussion on problem domain based on published articles

Understanding a few important topics is necessary in order to discuss the issue area of chicken disease identification employing AI, particularly through a web application: the difficulties in managing poultry diseases, the role that AI plays in resolving these difficulties, and the particular factors to be taken into account when putting a web app into practice. Enhancing early diagnosis and illness management with AI-powered web app poultry disease detection offers great potential. More practical approaches to developing a user-friendly online application, utilizing AI's capabilities, and addressing the many chicken health concerns can all help poultry producers find more affordable and efficient disease control solutions.

3.2 Discussion on problem solutions based on published articles

When talking about AI-based solutions for detecting chicken diseases in online applications, it's important to highlight the many methods that have been investigated and the useful outcomes that may be obtained from them. The integration of artificial intelligence (AI) with online applications for the diagnosis of chicken diseases may greatly improve the precision and effectiveness of disease control, as demonstrated by published publications and case studies. Behavioral monitoring, predictive analytics, image-based diagnostics, and interaction with veterinary services are some of the solutions. The acceptance and efficacy of these technologies are further supported by an emphasis on user-focused design, confidentiality of information, and security. By putting these suggestions into practice, one may eventually increase the production and overall health of chickens by improving management techniques and early illness identification.

3.3 Comparison of leading solutions

Poultry AI

Overview: Poultry AI is an artificial intelligence platform that employs picture analysis to identify illnesses in hens. It analyzes hen photos and detects illness symptoms using deep learning algorithms.

Principal Elements:

- Web-based program with an emphasis on picture interpretation.
- Uses physical symptom analysis to identify a variety of poultry illnesses.
- Based on AI analysis, offers suggestions and diagnostic assistance.
- Updates the illness database on a regular basis when new information becomes available.

Poultry Disease Detector

Overview: By examining user-provided photos and symptoms, this online application uses artificial intelligence to diagnose diseases in chicken. Its goal is to assist farmers in promptly detecting and controlling infections in their flocks.

Important characteristics:

- An intuitive online platform for posting chicken photos.
- Artificial intelligence algorithms that are taught to identify common poultry illness signs.
- Offers advice on management and comments on possible illnesses.
- Connection with an archive of data to get information about diseases and available treatments.

3.4 Recommended Approach

Table 1: Modules descriptions

Actuators	Functions
User	• Choose Images • Detection system.

CHAPTER 4

Methodology

4.1 What to Use

For chicken producers and veterinarians, an AI-powered online app for detecting diseases in chickens can be a useful resource. let's people submit pictures of sick hens that they have taken. These photos are subjected to AI models, which identify symptoms of illnesses including "Coccidiosis," "New Castle disease," "Salmonella," and "Healthy." In this chapter, I'll discuss how to achieve a project's goal using the detection of chicken sickness. Using deep learning models to get the accuracy which made a website to create a website using JavaScript, Django for detect exact chicken disease. An effective solution for improving poultry management of health is an AI-powered online app for illness identification in chickens. An app like this can greatly increase the efficacy and efficiency of managing diseases in chickens by providing early illness diagnosis, behavioral monitoring, predictive analytics, and interaction with veterinary services. A comprehensive framework for development, design, etc., known as the SDLC life cycle paradigm is accepted as the appropriate method. I am knowledgeable with many SDLC model types. The Big Bang, the Spiral, a waterfall, Agile, Iterative, and Adaptive System Construction models are a few of the paradigms used in software development. Each model provides a framework to help guide the development and application of the vehicle

elements platform. The particular needs of the SDLC model will tailor the veterinary development domain in order to build an efficient development process that is in accordance with the objectives of the disease detection through coop of the chicken of this project.

4.2 Why to use

The system architecture has to be defined as the initial stage in the development process. Determining the parts and their interrelationships was a portion of this. The system network layout placed a high priority on security, dependability, and scalability. This included detaching the database administration and back-end operations of the product from its user interface. Additionally, security measures were built within the design to guarantee safe transactions and safeguard computer data. For every software project, the agile methodology must be applied. I am acquainted with several of the terms used to describe agile approaches, including feature-driven development, scrum, quartz, flexible system development technique, and kanban. Nevertheless, I used the DSDM technique to achieve my goal. The DSDM approach is advantageous for a number of reasons. For iterative development, the dynamic system design approach is employed, which allows for flexibility in adjusting requirements. In situations where prompt delivery is necessary, this tactic works well. The use of an AI-powered web app for chicken illness diagnosis has several benefits, such as cost effectiveness, enhanced accuracy, early disease detection, and data-driven insights.

4.3 Section of methodology

To determine how to assess the data which was used throughout this investigation, a variety of techniques or approaches may be applied. A multi-step methodology that included model construction, expansion and improvement, data gathering, and manufacturing was employed in this study.

Pre-Project Phase:

- **Feasibility Study**: This stage entails assessing the infrastructure, financial, and functional viability of the project concept. It entails balancing the project's possible costs, advantages, and risks.

- **Conditions Collecting**: The prerequisites for the program have now been gathered and recorded. Determining the project's scope requires an understanding of business requirements, customer requests, and restrictions.

- **Planning:** Planning includes developing a strategy plan that details the project's goals, schedule, necessary resources, and deliverables. Determining the project's stakeholders, establishing roles and duties, and developing a partnership and risk management plan are all essential.

Project Lifecycle Phase:

- **Gathering Data:** I collected and analyzed online statistical data using Kaggle to create a trustworthy set of my own. A large, comprehensive dataset is not readily available in this sector because to the difficulties in locating and obtaining data for the several chicken disease conditions, the four stages' pictures of the different illnesses, and other considerations.
- **Data preparing:** After being collected in its data form from various vetenary sectors, each type of data was handled separately. Errors can occur in many data sets, particularly in noise. Technically speaking, I process the knowledge first, then go on to the next phase using the selected data set.
- **Data Preprocessing:** As every class was evaluated, the findings grew and became more focused. I had to adjust the size and add information for it to work. I restricted the amount of changes I made to the biggest and most appropriate ones since I was concerned about overfitting.
- **Model Selection:** To increase accuracy, train and assess the selected model using the available data after you've made your selection. DL uses a wide variety of

models. Using my technology, many versions of the concept were tested to determine the optimal configuration for precise data calculation.

- **Evaluation of Performance:** This section provides an explanation of each result. Our level of dependability for the next two courses was insufficient due to these tactics, even after testing and training. They designed a web-based tool for identifying various chicken diseases and produced visuals for f1 measurement, recall, efficiency, and confusion matrix.
- **Design:** This step involves building the software design using the gathered requirements. The high-level and extensive design activities it entails include database, architectural, and user interface design, to name just a few.
- **Development**: This stage involves coding the software using the design criteria as a reference. Developer writes the source code, run unit tests, and combine components to create a working software product.
- **Testing**: The goal of this step is to ensure that the software is high-quality and functional by putting it through its paces. It covers a wide range of testing methods, such as unit, user legitimacy, integration, and system testing.
- **Deployment**: The program is put into use once it has been approved and put through a rigorous testing phase. The appropriate environment must be used for installation, configuration, and setup of the software.

Post-Project Phase:

- **Maintenance**: After deployment, the program enters the maintenance phase. This phase involves ongoing maintenance, bug fixes, and updates to ensure the application continues to work and meets changing needs.
- **Evaluation:** The effectiveness of the project may be judged by contrasting its actual outcomes with its intended objectives. It helps identify areas that require modification and lessons learned for subsequent initiatives.

- **Closure:** At this point, the project formally concludes. It comprises finishing the project's documentation, keeping track of its artifacts, and doing a post-mortem or assessment of the project.

From initial planning to support after deployment, these sections provide a structured approach to managing projects involving software development and help achieve successful outcomes.

4.4 Implementations plans

The completed application is now made usable by the general public at this stage of the project. The new system has to be activated as soon as a bug is found and repaired. The settings, procedures, and release requirements are chosen in this section. If all goes according to plan, the updated system is then tested and implemented. To ensure accuracy, the data collecting must be provided after all further procedures have been finished. I broke the task down into its most crucial parts to make it simpler to complete. To make sure my work is done correctly, I have to follow these rules.

- Collections of datasets.
- Actions taken before processing an image
- Predicting class pictures for four chicken illnesses.
- The several algorithms in use.
- Using Resnet to categorize photos of chicken diseases
- Evaluate the accuracy and outcomes.

CHAPTER 5

Planning

5.1 Project Plan

I used tools like Kaggle to compile all of my datasets publicly. I selected a dataset that seemed to make sense for the different chicken coop conditions. I could then start working on preparing the data after that. I started tinkering with the programming before trying the idea out. I assessed each of the four employed algorithms' accuracy. I considered the accuracy and selected the one that would work best for my needs. Based on data on conditions in chicken coops, this has proven to be fairly reliable in identifying diseases in chickens. A set of basic criteria has been constructed after a careful analysis of all relevant mathematical and philosophical ideas and techniques. These requirements are necessary for each and every attempt at photo categorization made with the build web app tool. Prior to development, every project has to have its possibilities, cost, schedule, management of risks, and interaction server protocols established. In order to minimize risks that might compromise the developer's capacity to finish the project, planning is essential before it is begun. Project planning can contain several aspects such as setting goals and objectives, controlling risks, meeting deadlines, and so on. Project planning standard tools include time boxes, these are often used in software project plans. These technologies are essential for carefully arranging activities related to intuitive illness detection in chickens in a systematic way.

5.1.1 Management plan

Explain the roles and responsibilities of the project team and the project management process. Establish the channels of communication and reporting to ensure a successful collaboration. Establish the decision-making and escalation procedures stages of the issue-resolution approach.

Table 2: Management Planning

No	Task Name	Duration	Start Date	End Date
1	Introduction	5	31 July 2023	4 august 2023
2	Initial Study	4	10 august 2023	13 august 2023
3	Literature Review	4	19 august 2023	22 august 2023
4	Methodology	3	25 august 2023	27 august 2023
5	Planning	10	1 September 2023	10 September 2023
6	Feasibility	15	15 September 2023	30 September 2023
7	Foundation	5	3 October 2023	8 October 2023
8	Exploration	14	13 October 2023	27 October 2023
9	Engineering	30	29 October 2023	28 November 2023
10	Deployment	18	13 December 2023	31 December 2023
11	Testing	10	5 January 2024	15 January 2024
12	Implementation	15	20 January 2024	5 February 2024
13	Critical Appraisal and Evaluation	4	10 February 2024	14 February 2024
14	Lessons Learning	3	17 February 2024	19 February2024
15	Conclusion	1	21 February 2024	22 February 2024
	Total	141 days		

5.1.2 Resource Allocation

Ascertain all of the project's assets, including personnel, equipment, and software. Determine the appropriate resource allocation based on the project timeline and workload. Assign team members responsibilities and tasks, and ensure they have the necessary skills and expertise.

Table 3: Resource Allocation

No	Task Name	Duration	Resource
1	Introduction	5 days	End User
2	Initial Study	4 days	Analyst
3	Literature Review	4 days	Analyst
4	Methodology	3 days	Analyst
5	Planning	10 days	Analyst, Designer, Developer
6	Feasibility	15 days	Analyst
7	Foundation	5 days	Designer
8	Exploration	14 days	Designer , Developer
9	Engineering	30 days	Developer
10	Deployment	18 days	Analyst, Developer
11	Testing	10 days	Analyst, Developer, Tester, Users
12	Implementation	15 days	Analyst, Developer
13	Critical Appraisal and Evaluation	4 days	Analyst, Tester and Developer
14	Lessons Learning	3 days	Analyst, Users
15	Conclusion	1 day	Analyst
	Total	141 days	

5.1.3 Time Boxing

Break up the project into several time periods or iterations to facilitate development and testing. Establish how long each time box will last, in addition to the tasks and products that need to be completed for each iteration. Set clear goals and provide resources for each time box.

Table 4: Time Boxing

Time -Box	Task Name	Duration	Resource
TB1	Introduction	5 days	End Users, Analyst
	Initial Study	4 days	Analyst
	Literature Review	4 days	Analyst
TB2	Methodology	3 days	Analyst
	Planning	10 days	Analyst, Designer, Developer
	Feasibility	15 days	Analyst
TB3	Foundation	5 days	Designer
TB4	Exploration	14 days	Designer, Developer
	Engineering	30 days	Developer
TB5	Deployment	18 days	Analyst, Developer
	Testing	10 days	Analyst, Developer, Tester, Users
TB6	Implementation	15 days	Analyst, Developer
TB7	Critical Appraisal and Evaluation	4 days	Analyst, Tester and Developer
	Lessons Learning	3 days	Analyst, Users
TB8	Conclusion	1 days	Analyst
	Total	141 days	

CHAPTER 6

Feasibility

6.1 All possible types of feasibility

6.1.1 Operational feasibility

A feasibility study determines the likelihood that all relevant factors, including engineering, organizing, legal, and financial concerns, will be taken into account in order for a project to be successfully finished. Operational practicability is a degree that shows how well a system ages, makes use of the scope established during opportunity definition, and meets the requirements established during project or requirement analysis phase of system development. We have designed a web application that will allow to detection specific chicken disease using deep learning, step-by-step way. The proposed idea is centered around a web application that serves as a chicken disease sing coop of chicken to detect class of disease.

6.1.2 Technical feasibility

Hardware	Software
Dell Laptop, Wi-Fi, Router, Cable, Android Phone	Android Studio, Google Chrome Browser, Windows, MS Word, VS code

6.1.3 Technology

Client slide	Server side
Html, CSS, JavaScript	Django/Python

6.2 Cost Benefit Analysis

Project managers use the concept of cost-benefit analysis to examine the benefits and downsides of different project routes, including interactions, activities, business needs, and investments. Out of all the options available, a cost-benefit analysis shows me the best course of action to accomplish my goal for the least amount of money.

Project Name: Chicken disease detection web app using AI

Table 5: Cost Benefit

Equipment	1st Year	2nd Year	3rd Year	4th Year	Total
Web Based Application	20000				20000
Domain & Hosting		10000	10000	10000	30000
Software	1000				1000
Internet	2000	2000	2000	2000	8000
Model Training	5000				5000
Development		5000			5000
Maintenance	10000	10000	10000	10000	4000
Total					73,000 BDT.

6.3 DSDM Dynamic System Development Method (DSDM)

The Dynamic Systems Development Method, or DSDM for short, is an agile project and software development management organizational framework rather than a specific tool or technology for creating applications. It places a strong emphasis on iterative development techniques, frequent delivery of working software, and collaboration

between teams of developers and business stakeholders. It is important to keep in mind that using certain tools or technologies—like JavaScript, Html, CSS with Django python—is required by DSDM. These are commonly-used web development technologies that have definite use in DSDM projects.

CHAPTER 7
Foundation

7.1 Some potential approaches

7.1.1 Interview

During any project, the timing of interviews is vital. In order to help users use the chicken disease detection system to solve employing coop photographs of chicken illness, I am able to gather information on chicken disease, listen to users' demands, and connect with qualified individuals by conducting interviews. Interviewing people can provide me with a wealth of knowledge on detecting chicken sickness. By taking part in interviews, I may discover all the needs as well as any communication obstacles with resource allocation.

7.1.2 Observation

With the aid of a single platform may complete a variety of paperwork both before and after their diseases detection by means of the proposed project, which is a web application designed to detection system of disease. We have devised an online application that enables to detect chicken disease efficiently and quickly handle every aspect of their detection system in a detailed, step-by-step fashion.

7.1.3 Data Collection

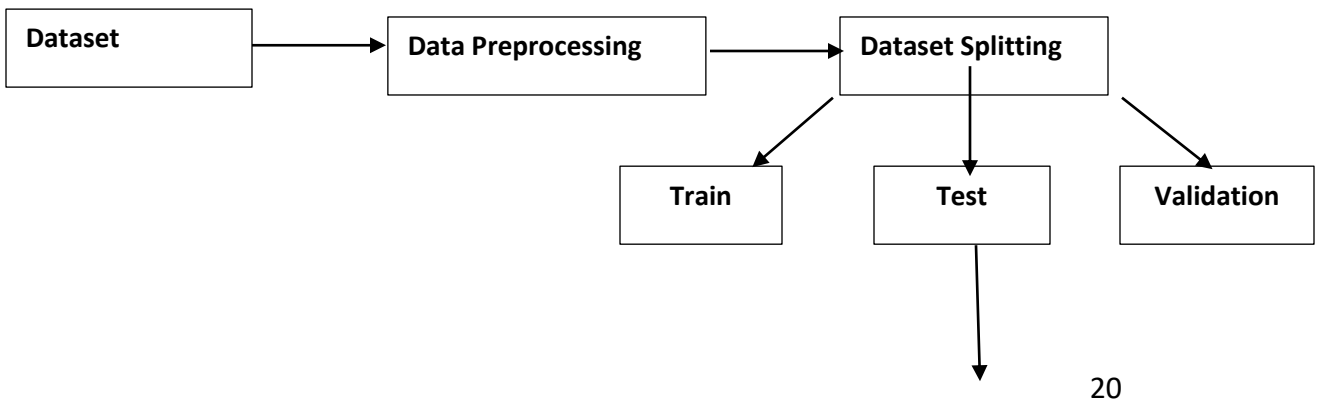
I have gathered a total of 8067 pictures. The online data that were collected were given via Kaggle sources. The dataset's four groups—"Salmonella," "Coccidiosis," "New Castle disease," and "Healthy"—are called after the kind of chicken illnesses they are representative of. Twenty percent of the data are designated as train, while the remaining fifty percent are used for test and validation.

Fig 1: Sample of dataset

7.1.4 Data Processing

Every image in the gathering I utilized was created by combining the different online field data that Kaggle had gathered about height and breadth. For my model, each image needs to fulfill specific quality requirements, therefore I changed the script to make the image 224 by 224 pixels. In addition, before processing every picture in my model, I prefixed it with "jpg". I separated and altered the photos after data augmentation in order to get them ready for categorization. As a result, I used the split counterpart of every dataset to train the framework.

- Images with predetermined sizes based on codes.
- The file types will be converted to JPG.
- Get rid of any incorrect images.
- Removed pointless pictures



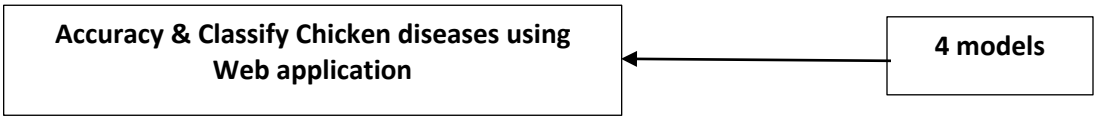


Fig 2: The recommended model for the whole research project.

7.2 Specific problem are identification and description

During the process of creating and deploying an AI-powered online application for detecting poultry diseases, certain issues and their explanations surface in many crucial domains. Determining and fixing these issues is essential to developing a useful and efficient application. Creating an AI-powered online application for detecting chicken diseases involves a number of difficulties, such as problems with data integrity, model abstraction, immediate processing, unity, privacy of data, model servicing, cost, ethical considerations, and educating users. Careful planning, ongoing enhancement, and support are needed to address these particular issues and guarantee that the app is dependable, efficient, and easy to use.

7.3 Possible solution

A mix of technological, pragmatic, and strategic solutions may be used to handle the particular issues with an AI-based web app for detecting chicken sickness. 4 types of classes have been used for detect using chicken coop of following models of deep learning. Using AI to address the problems with a web app for detecting chicken disease requires putting solutions in place for a number of issues. By implementing these fixes, app developers may raise the app's efficacy, dependability, and user happiness, which will improve disease control and result in healthier flocks of chickens.

7.4 Overall Requirement List

- Functional Requirements
- Non-Functional Requirements.

7.4.1 Functional Requirements

7.4.1.1 User

- Detection system using DL.

7.4.2 Non-Functional Requirements

7.4.2.1 Performance

Performance have been good after gathering good accuracy for prediction specific disease of chicken.

7.4.2.2 Availability

To use the system at any time and from any location, users only need a PC with an Internet connection. The system works with a number of web browsers, including Internet Explorer, Mozilla, Opera, and Chrome.

7.4.2.3 User Friendly

The technology is user-friendly and boasts an intriguing UI.

- There should be no appreciable lag or downtime when a big number of users are using the website simultaneously.
- The website has to be capable of managing substantial volumes of data.
- The design and style of the web page must be simple and easy to use.
- New functions and features must be added to the website with ease, without needing a lot of reworking or rewriting.
- It is necessary to maintain the website in order to address bugs and post-deployment difficulties.

7.5 Which technology to be implemented

The program I'm working on is entirely web based. I'm developing my project with JavaScript, Html, CSS, Django, Python.

HTML: Markup languages such as HTML may be used to construct web pages. Web browsers can view and comprehend HTML files. HTML components are the foundation of every website. permits the use of graphics and HTML components, allowing you to

produce interesting content. It can also generate titles, links, chapters, lists, and quotations in [2].

CSS: We may use CSS to change the fonts, colors, and layouts to further customize the content on our homepage. This enhances our website's visual appeal and coherence. As a result, the website appears more inviting. Within [2]

JavaScript: Right now, one of the most widely used programming languages is JavaScript. As a language for web development, we also use JavaScript. This develops a layer of standard web technologies. In [3]

Bootstrap 4: As of right now, Bootstrap is version 4.0. Using Bootstrap 4, you can create responsive websites with all the necessary HTML, CSS, and JavaScript components. I consequently granted users accounts on my website.[4]

Django: Django is an advanced Python web toolkit that promotes efficient development and simple, straightforward design. Because of its comprehensive feature set, scalability, and ease of use, it is a popular choice for developing applications for the internet.

A researched topic is an area of research that is currently being looked at and analyzed to clarify ideas for developing models, setting and achieving goals, gathering data, managing, teaching, and improving performance. I go over the equipment and techniques I use to take measurements. Python programming, Microsoft software, and other tools like Scikit-learn were all made possible by NumPy. The facilities of Google Co Labs are only used for teaching and evaluation. Complex DL algorithms and Python-based data analysis may be written by Google Colab programmers.

Libraries:

- **Matplotlib:** Py-plot graphing is a collection of techniques that is part of Matplotlib's visualization capabilities. Just two of its many uses include drawing borders and establishing lines inside a plot. Shapes are another.
- **NumPy**: Using the NumPy library to work with matrices in Python is one common method. This section covers the Fourier transform, matrix operations,

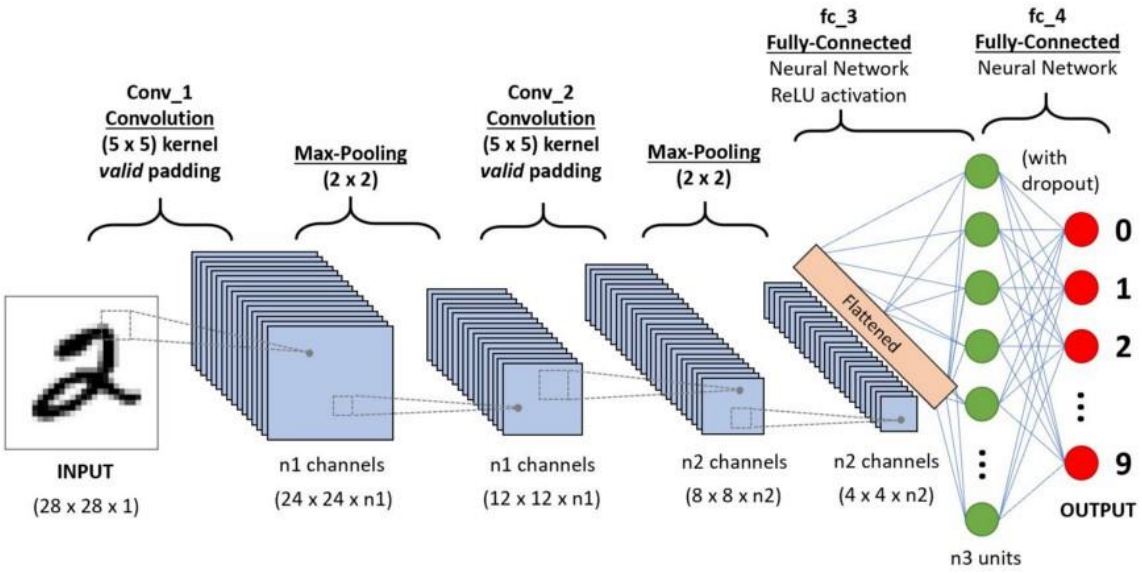
and the foundations of algebraic geometry. Assets and instruments in Python are provided by the NumPy library for handling matrices of different sizes. NumPy makes it possible to build arrays precisely and methodically. Numerical computations are performed using the NumPy Python library. The term "a wide range of varieties Python" is also used.

- **Scikit-learn**: This is a helpful and user-friendly forecasting and data analysis application. Open-source software is freely accessible for any individual to use and alter to their own specifications. Through growth, Matplotlib, SciPy, and NumPy were utilized.
- **Seaborn:** This well-liked data visualization tool, which has a long history of integrating with matplotlib, is easy to use and offers a straightforward tool for producing visually appealing and captivating data visualizations.
- **CV2**: A collection of Python extensions known as Python's OpenCV was designed to address problems specifically related to computer vision. It may also be used to identify objects and individuals by looking at images and videos, as well as notes made on the scene.
- **Job-lib:** Another more effective way to avoid repeating the same computation, saving time and money.
- **H5 Py:** The h5py module offers a Python HDF5 native data wrapper. Massive volumes of numerical data can be easily handled and stored with HDF5, thanks to NumPy's support.
- **OS:** The OS portion of the Python programming language offers a variety of tools that artists may utilize to communicate with the software that drives certain components of the computer equipment they work on.
- **TensorFlow:** A free Python math library that facilitates the development of deep neural network systems and self-learning methods.

7.5.1 Analysis

1. **CNN:** An example of a deep learning model is a convolutional neural network (CNN), which is mostly employed for the analysis of visual data such as

photographs. They are made up of many layers: fully connected layers that categorize the extracted features, pooling layers that downsample the input to minimize dimensionality, and convolutional layers that apply filters to identify features. An picture changes into a variety of feature maps as it travels through the network, emphasizing important patterns that are subsequently utilized for precise categorization. CNNs are able to distinguish objects and structures in pictures with accuracy thanks to its design.



2. **VGG16:** VGG16, the abbreviation for this model, is sometimes referred to as VGG Net. The method makes use of neural networks with sixteen-layer convolutions (CNNs). Considering that pretrained neural networks There are more than a million photos in ImageNet. A mouse, pencil, internet keyboard, and other items are among the 1000 unique item categories that the conditioned prior to use network can identify in images. To train the network, a wide variety of visually rich feature representations have been employed. The network may be able to handle the 224 × 224 image source size. For other pretrained network possibilities, view the Originally trained Parallel neural network models in MATLAB.

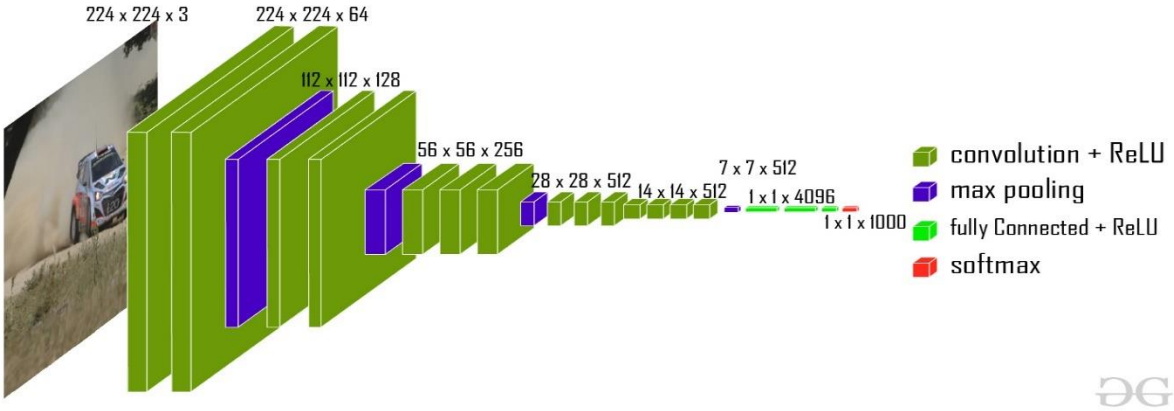
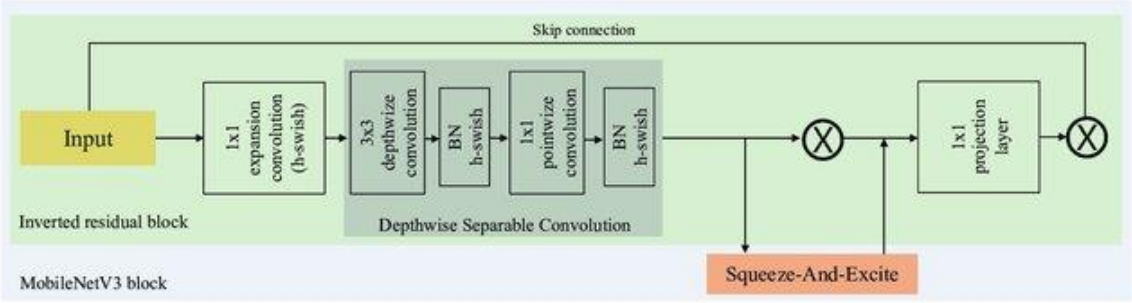


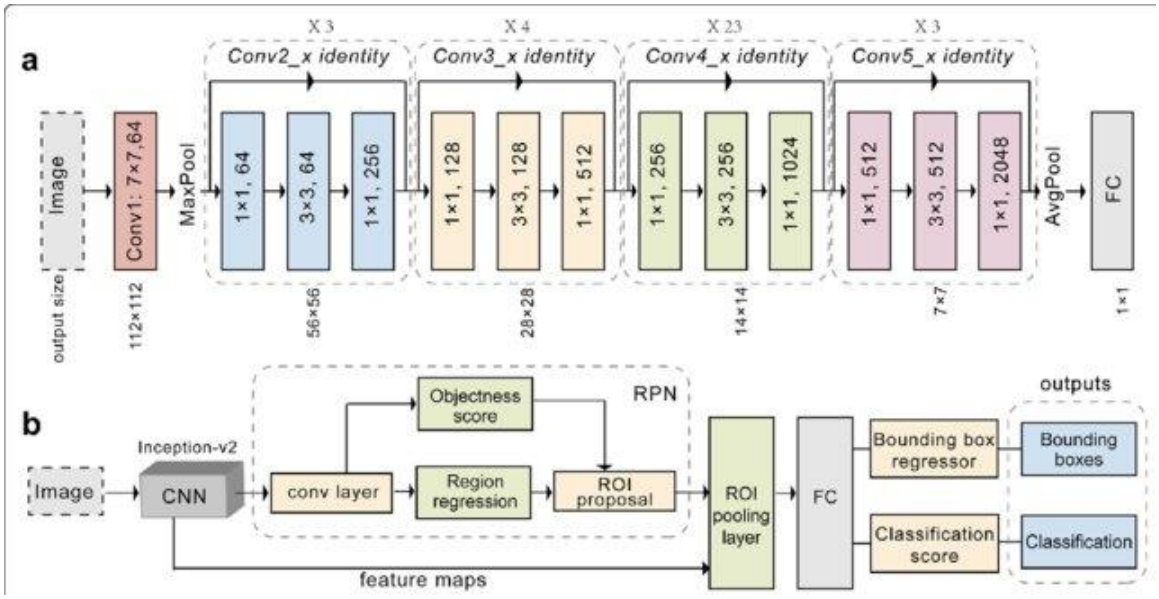
Fig : The architectural model of VGG16.

3. **MobileNetV3:** To reduce computing expenses, MobileNetV3 divides the convolution process into two efficient layers by applying depthwise separable convolutions to an input picture. It integrates attention techniques, such Squeeze-and-Excitation blocks, to focus on significant characteristics and employs lightweight activation functions, like Swish, for increased non-linearity. The feature maps are downsampled by pooling layers, and class predictions are generated by a fully connected layer by interpreting the extracted features. Because of its simplified design, MobileNetV3 can achieve great accuracy while using less resources, which makes it a good choice for edge and mobile applications.



4. **Resnet:** With the use of skip connections and a special design, ResNet, also known as Residual Network, enables input to bypass one or more levels and be added to the output. The vanishing gradient issue is lessened by this architecture,

which facilitates the training of extremely deep networks. The network is made up of many residual blocks with convolutional layers, batch normalization, and learning-enhancing activation functions thereafter in each block. ResNet can better perform in tasks like picture classification by capturing complex characteristics across several layers by allowing gradients to flow across these skip links. This architecture maintains training accuracy and efficiency while enabling the creation of incredibly deep models.



7.6 Recommendation and justifications

Using AI to implement a web app for detecting chicken diseases requires making strategic suggestions to meet a number of obstacles. The efficiency and usability of the software are improved by utilizing edge computing, learning via transfer, augmenting data, and user-centric design. A successful implementation is aided by integration with current systems, robust data protection safeguards, frequent model upgrades, thorough training, and ethical considerations. The app's creation and long-term viability are further supported by assessing cost-effectiveness and looking at financing options, which enhances poultry health management and efficiency of operation. To increase illness diagnosis accuracy, start with pre-trained algorithms on sizable picture datasets and refine

them using data unique to poultry. By utilizing pre-existing models that have acquired generic properties, transfer learning helps to minimize the quantity of data and training time required.

Python, Django, and JavaScript were utilized to build my project's front end and back end. By combining several technologies, the "Chicken disease detection web application using of AI" creates a comprehensive framework. The various components of this architecture apply the limitations mentioned for each system under investigation. There are several systems in each module. To ensure accuracy, set up mechanisms that will automatically retrain models using fresh data. Continue to track and assess model performance in order to identify and quickly resolve problems

CHAPTER 8

Exploration

8.1 Use case

This section uses use-case data and graphics to analyze both functional and non-functional demands.

User:

Following their login, the user can carry out the following tasks:

- Detection chicken disease through coop of chicken images.
- Basic Info about company
- Contact Us page

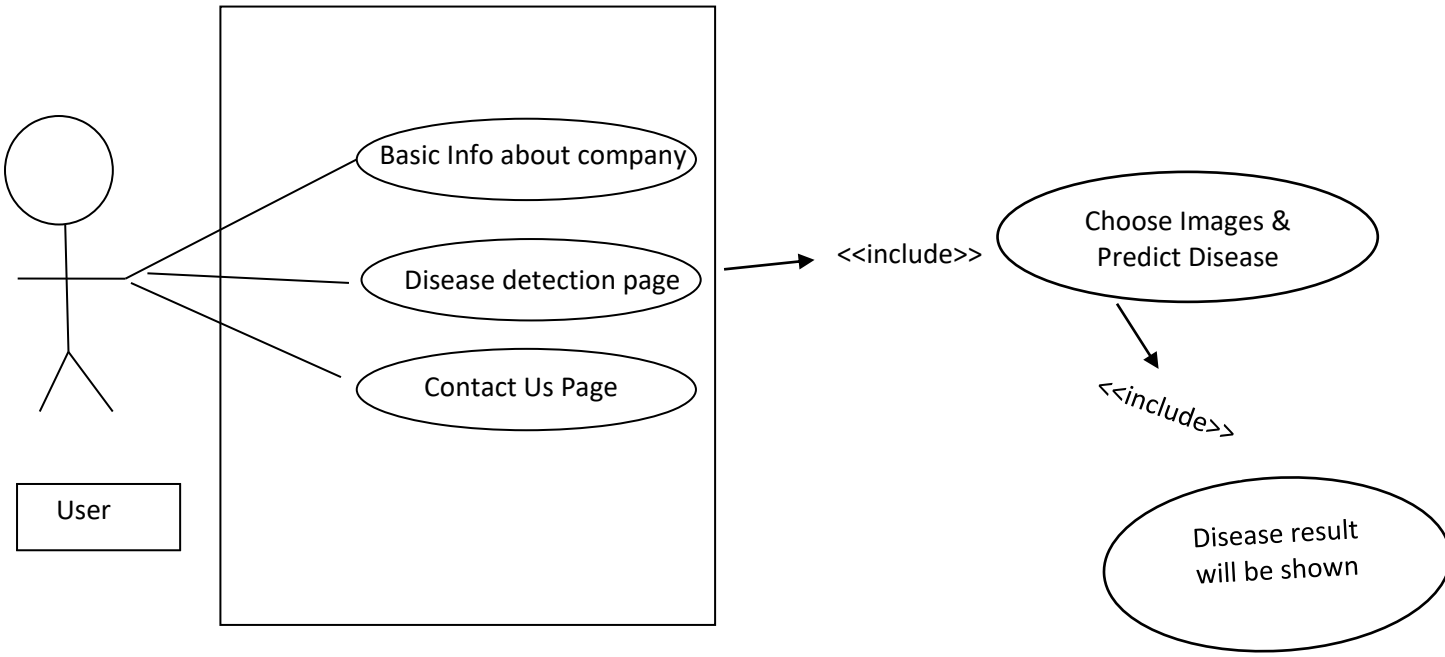


Fig 3: Use case Diagram

8.2 Activity diagram

Describe the dynamic aspects of the system. It looks like a flow chart that shows how one job is related to another. You might utilize the exercise to describe how the system works. Control is therefore distributed throughout activities. Each module's whole activity diagram looks like this: in command.

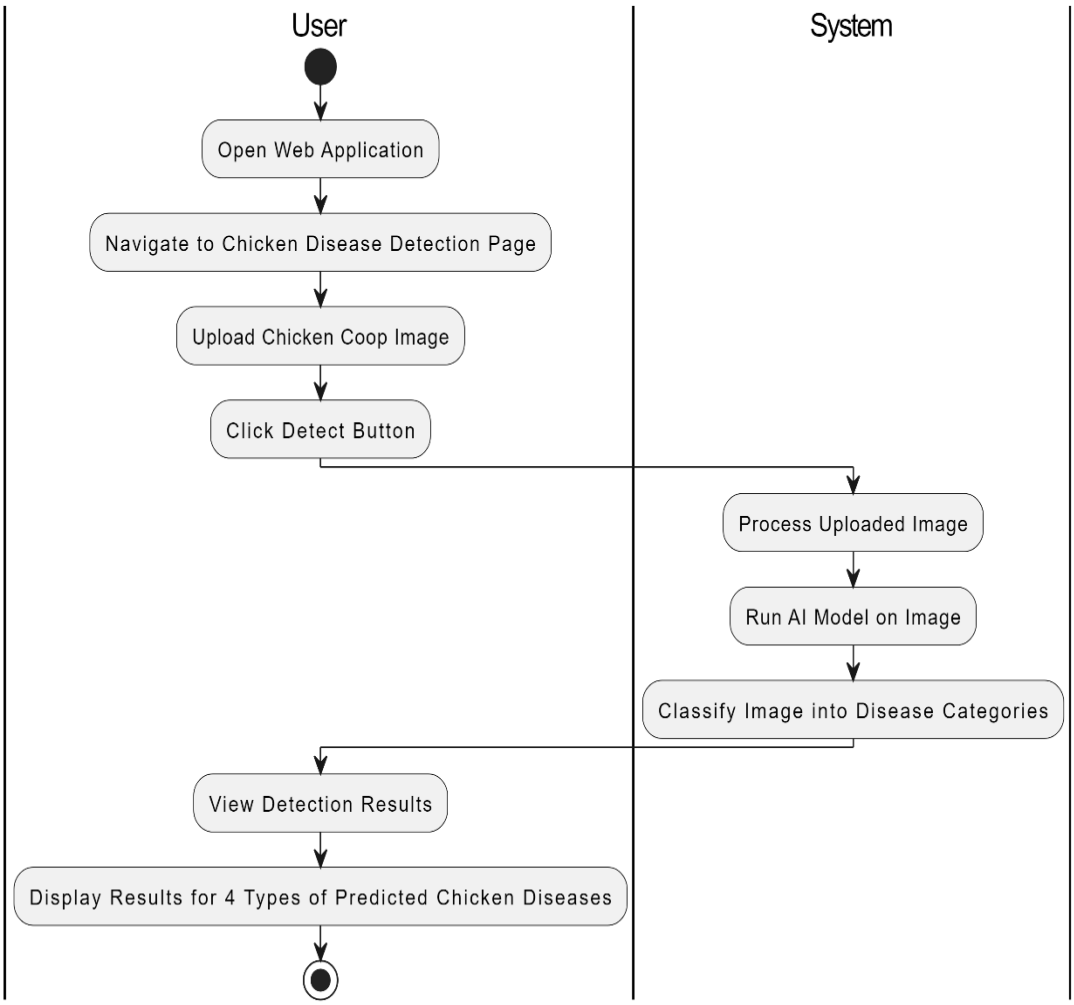


Fig 4: Activity Diagram.

8.3 Requirement catalogue

Functional requirements:

- FR1: Detection system using DL
- FR2: 4 models used and get the accuracy of models.

Non-Functional Requirements:

- **NFR1:** Records are easy to update and maintain.

- **NFR2:** To use the system at any time and from any location, users only need a PC with an Internet connection. The system works with a number of web browsers, including Internet Explorer, Mozilla, Opera, and Chrome.
- **NFR3:** The technology is user-friendly and boasts an intriguing UI.
- **NFR4:** View details of company and contact details.

User Interface Requirements:

- **UIR1:** Easy to use interface with clear navigational that makes it straightforward to access different features and capabilities.
- **UIR2**: Screen sizes and device sizes can be accommodated with responsive design.
- **UIR3:** Icons provide visual cues to aid in understanding and using the system.

Security and Privacy Requirements:

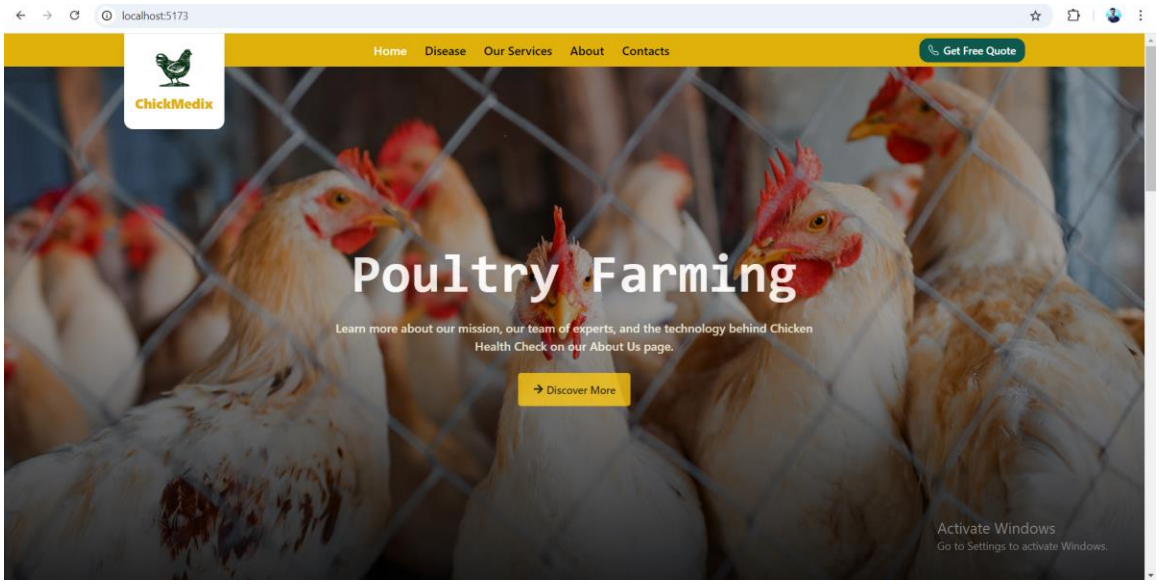
- **SR1:** Secure authorization and authentication protocols to protect user data.

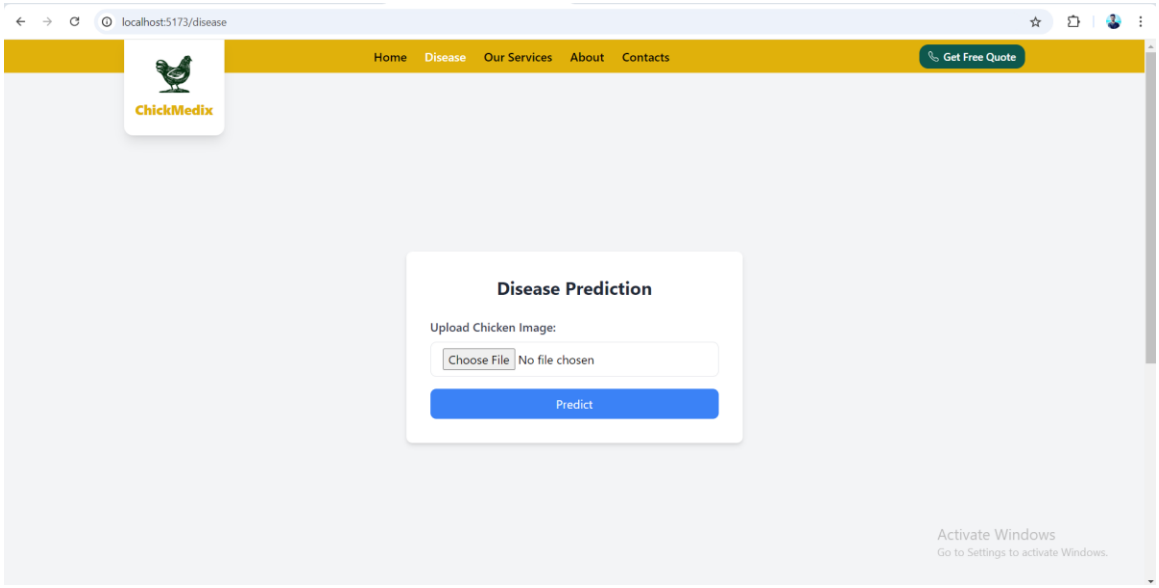
8.4 Prioritized Requirement List (PRL)

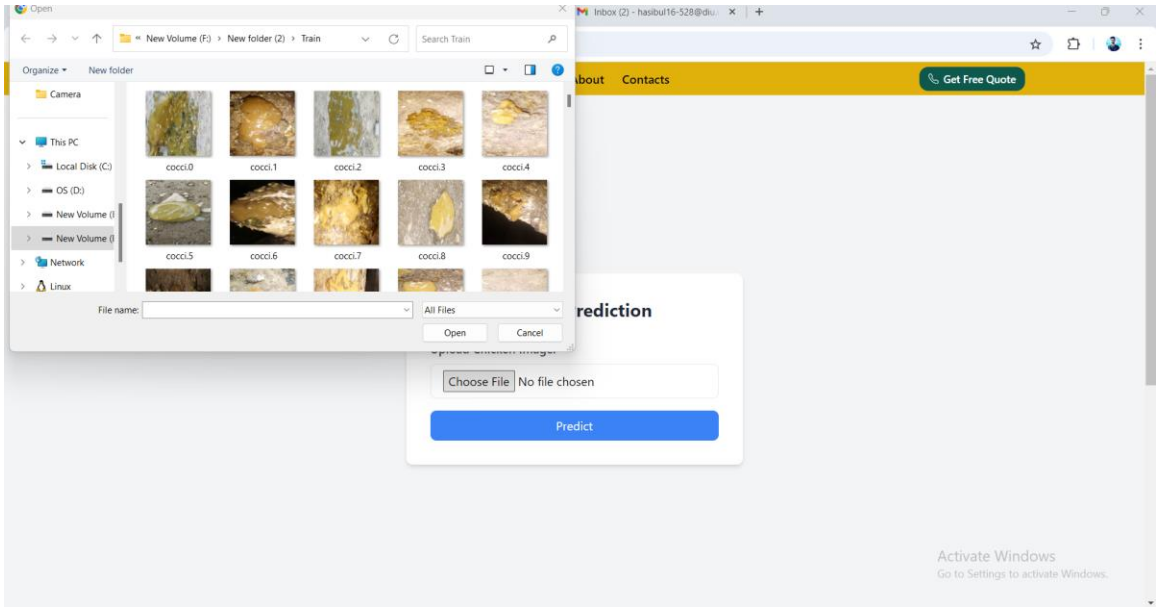
Table 6: Prioritized requirement list

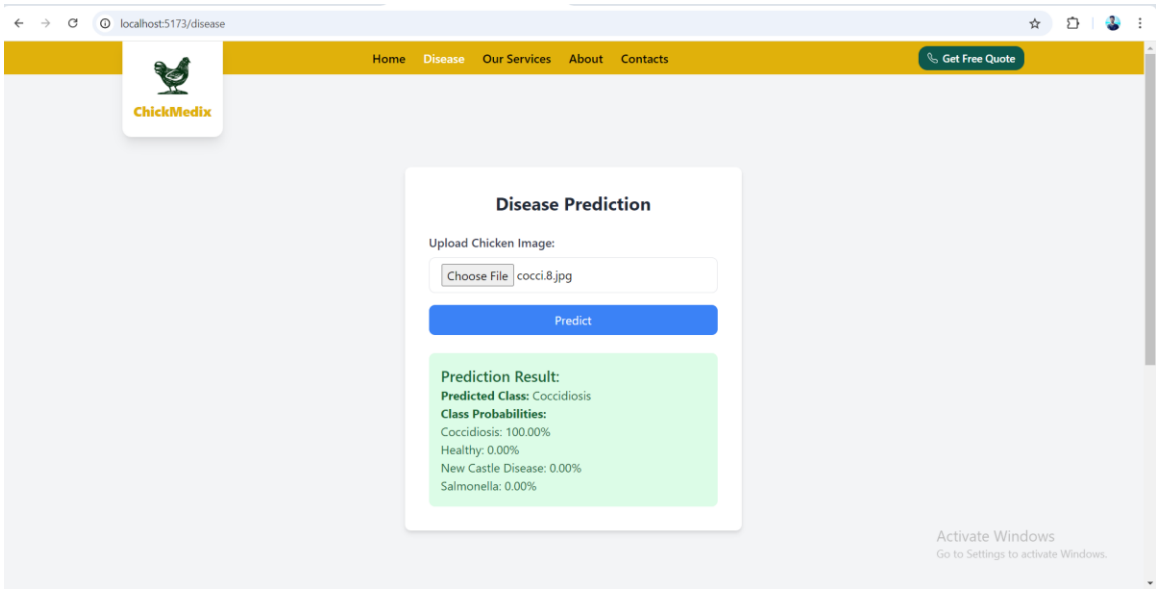
Requirement ID	Requirement Description	Priority	Dependencies	Status	Validation Criteria
RQ1	Detection of chicken system using DL models	High	RQ2		Detect 4 classes of chicken disease.
RQ2	Apply 4 DL models for web app & get accuracy	High			Give accuracy successfully.

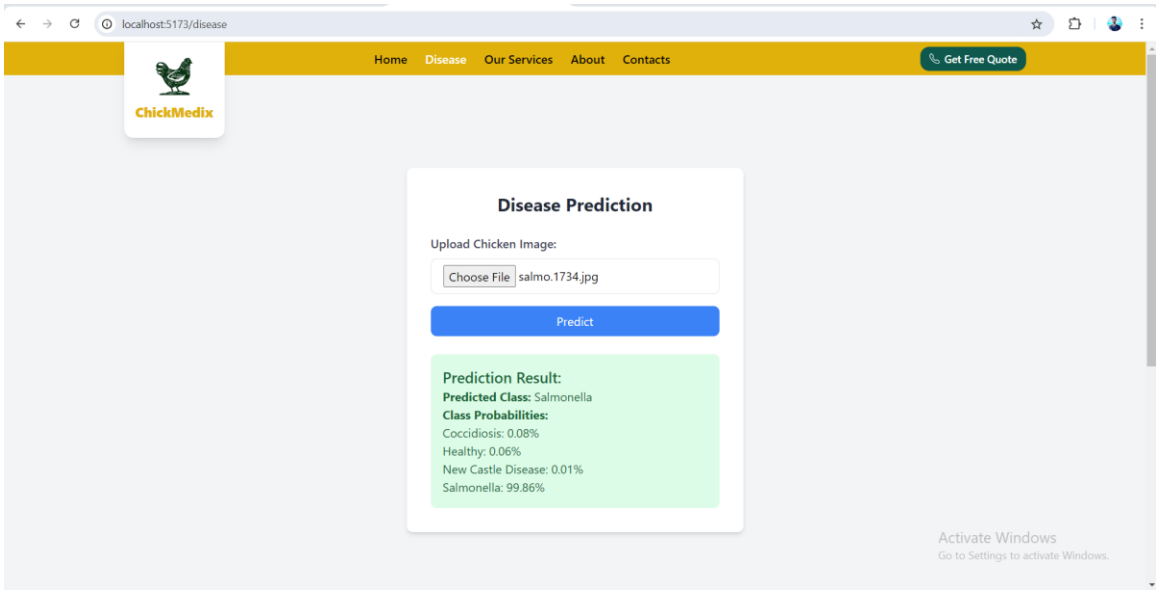
8.5 Prototype of new system

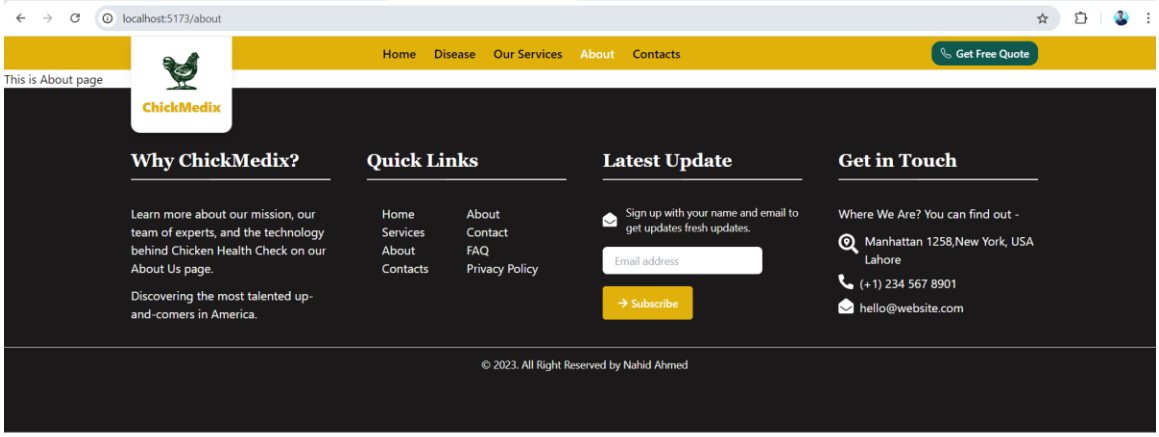












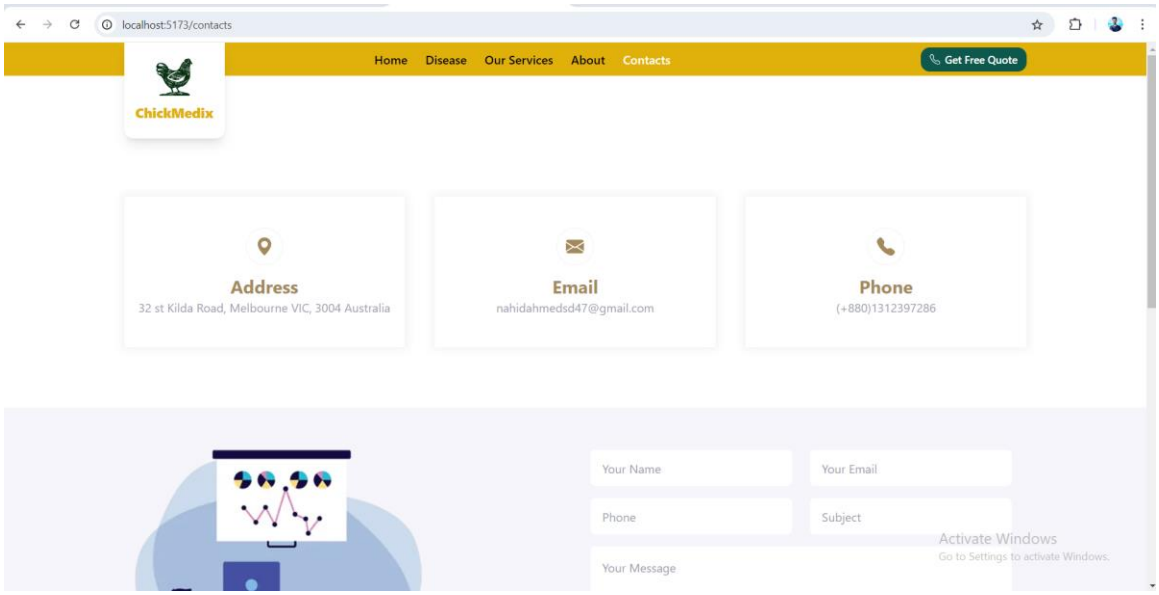


Fig 5: Interfaces for user dashboard account

CHAPTER 9

Engineering

9.1 Class Diagram

To display the original content for the interclass linkages, a class was built. Here, the class defines the variables and behaviors of an entity, either as a single definition of programming or as a distinct entity within a program.

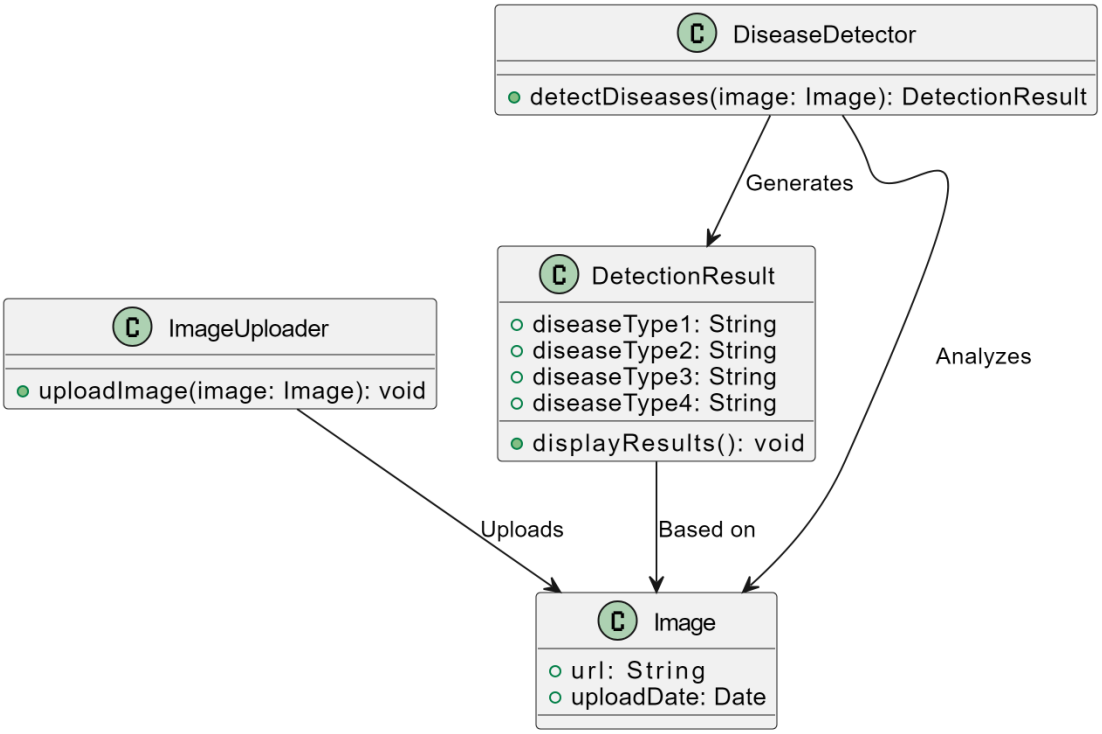


Fig 6: Class Diagram

9.2 ER diagram

A sort of structural application employed in design is institutional means of communication, often known as the ER model, ER Diagram, or ERD. The constrained system's primary components and the relationships between these entities are the two primary pieces of information that the ERD communicates and presents in different ways. The user module's entity-relationship diagram is now complete.

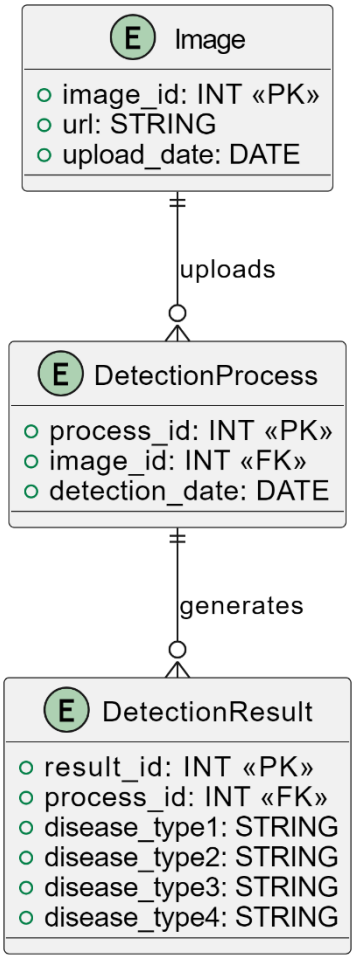


Fig 8: ER Diagram.

9.3 Sequence Diagram

This is all about one module's sequence diagram: User dashboard.

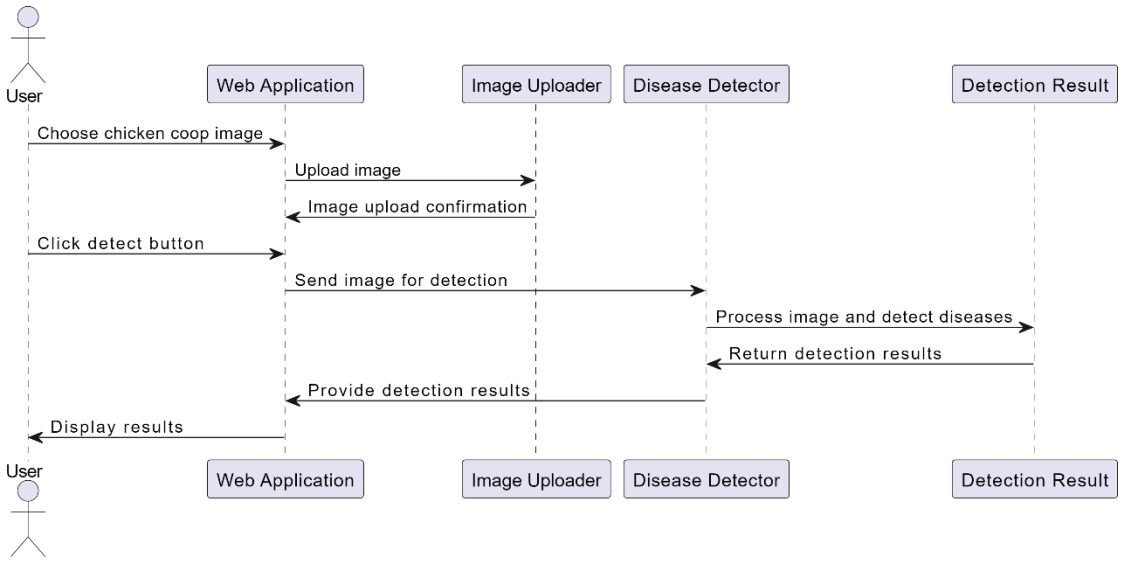
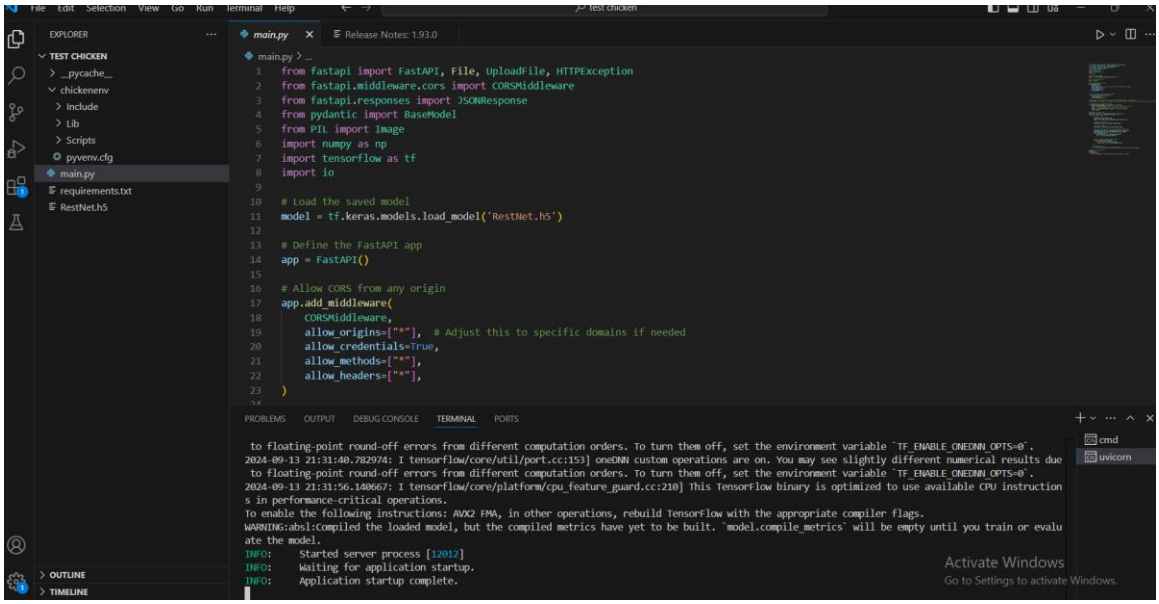


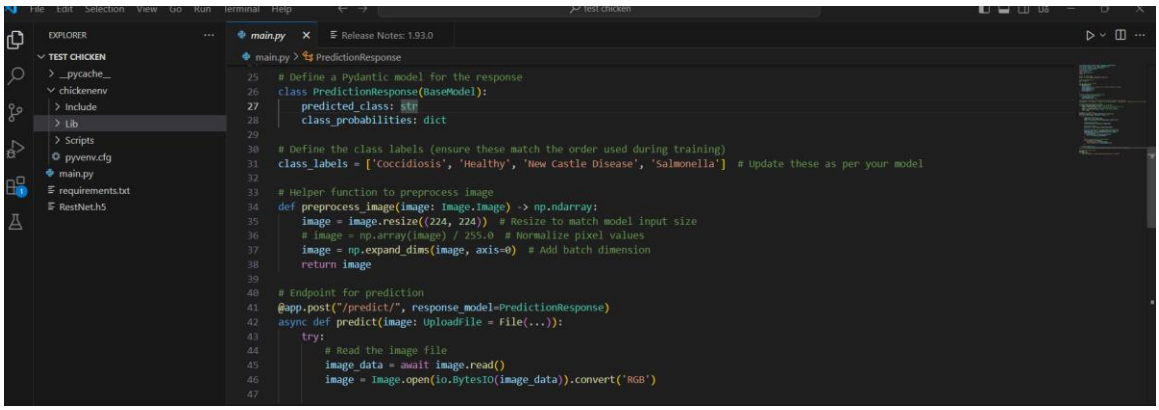
Fig 9: Sequence Diagram

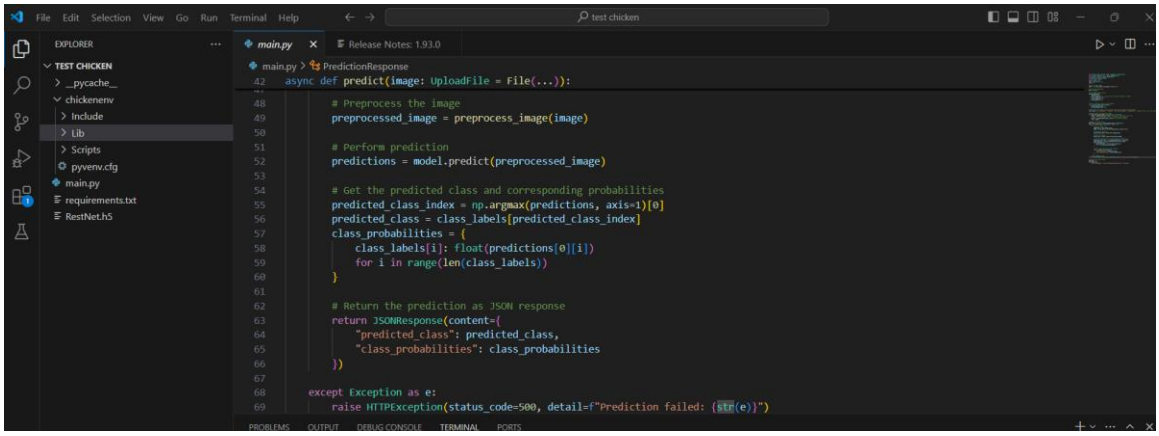
CHAPTER 10

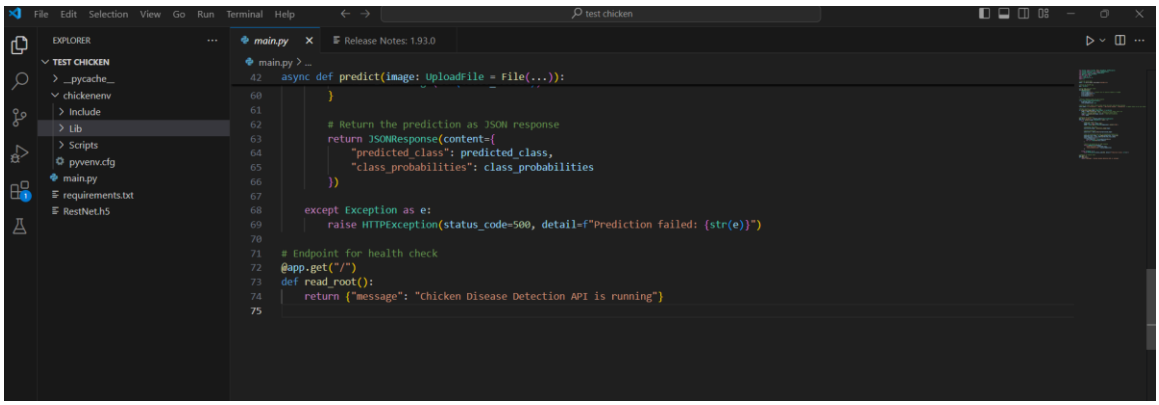
Development

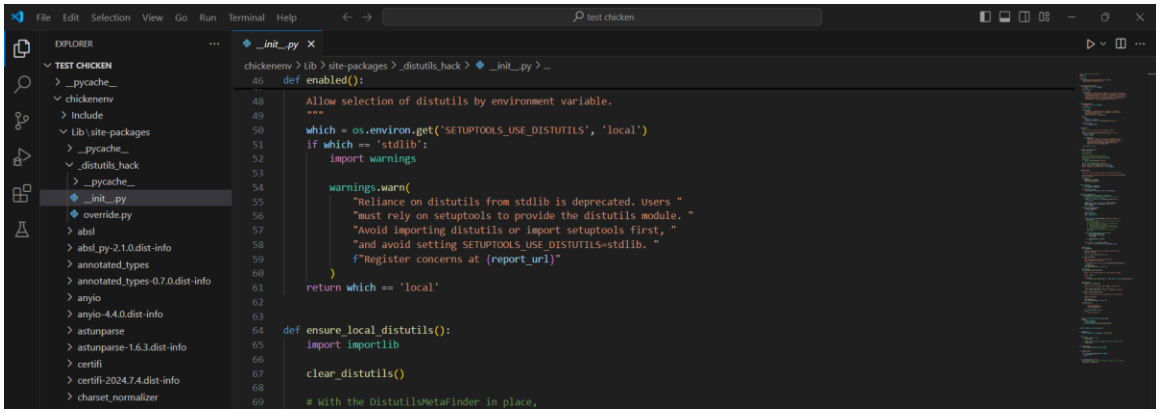
10.1 Core Module Samples

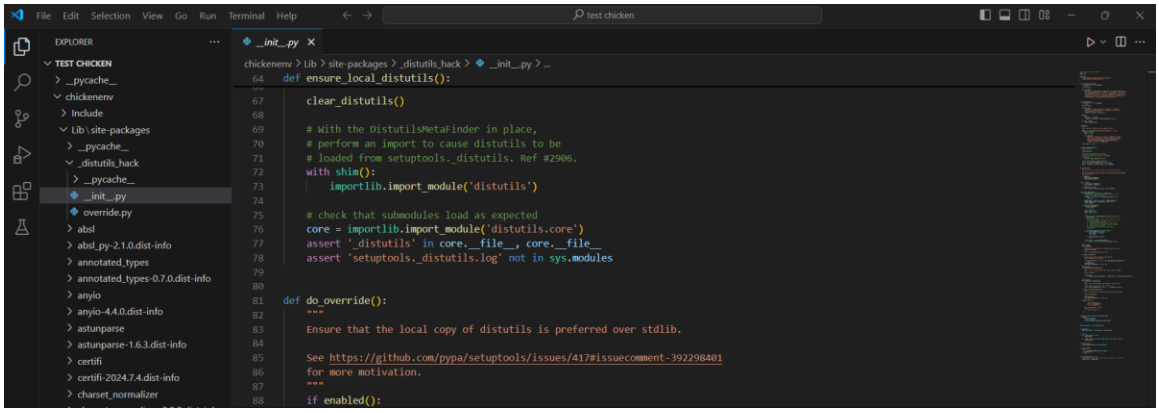


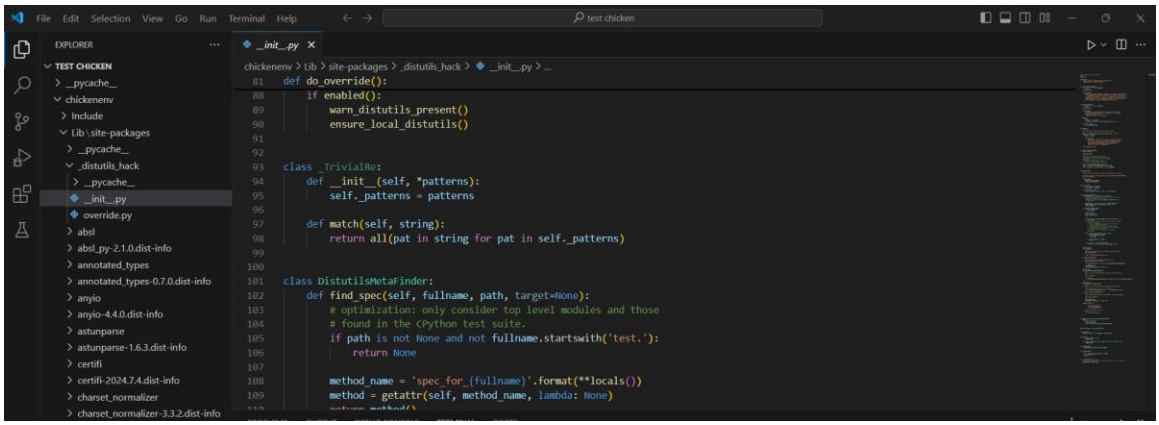


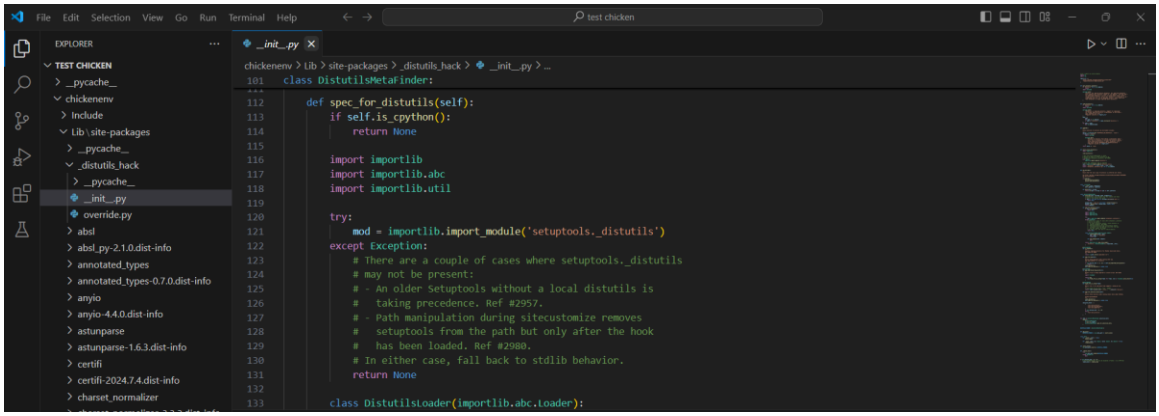


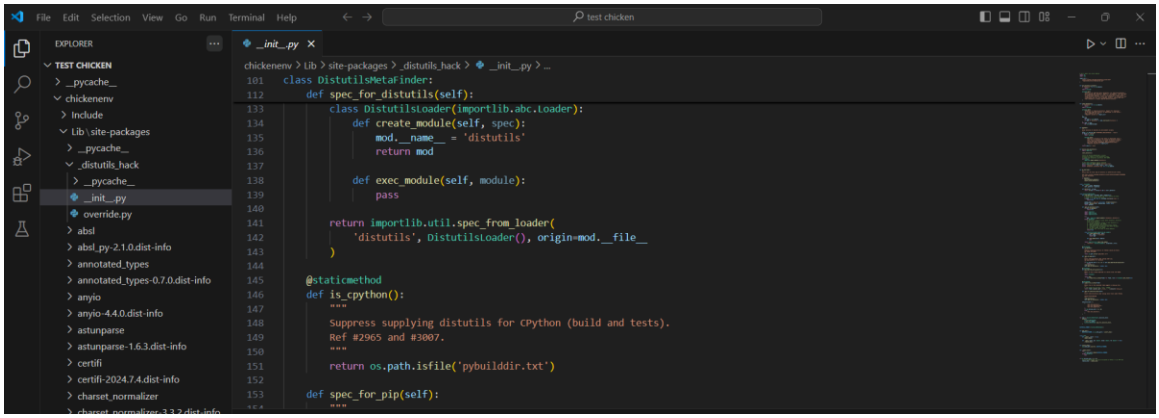


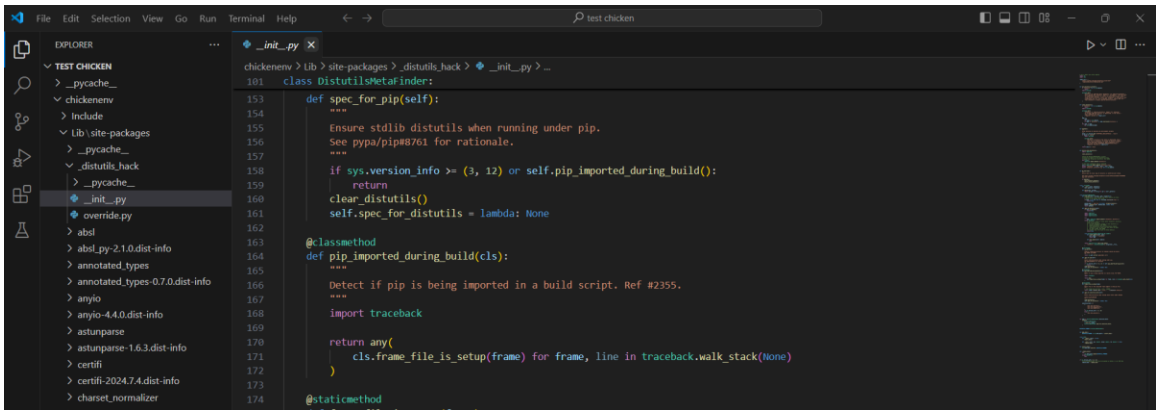


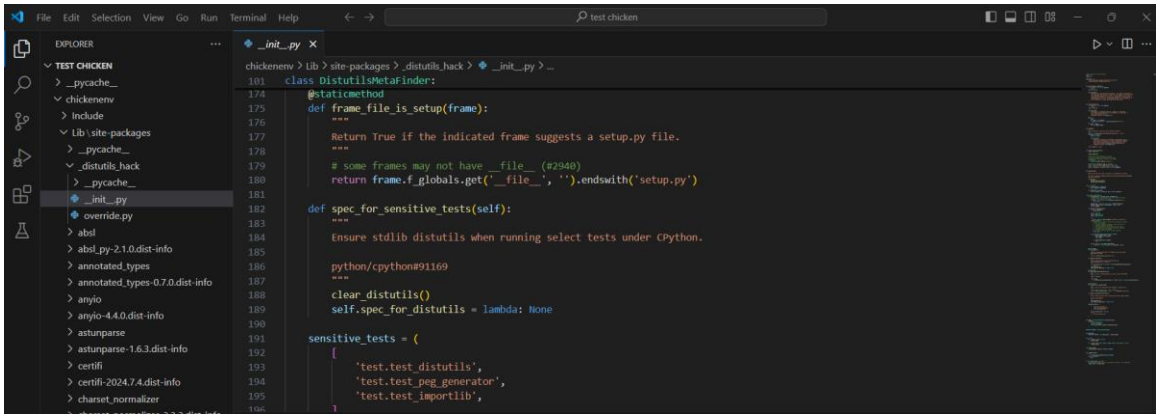


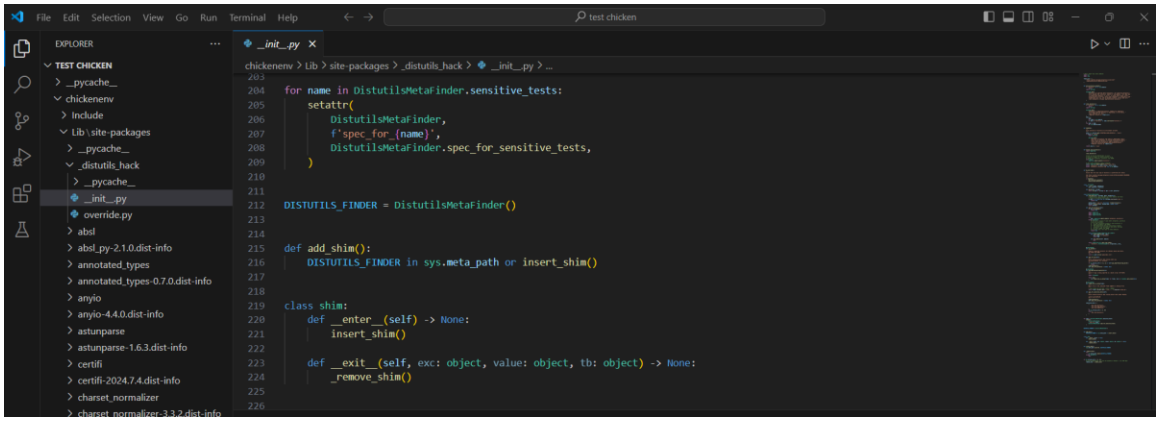


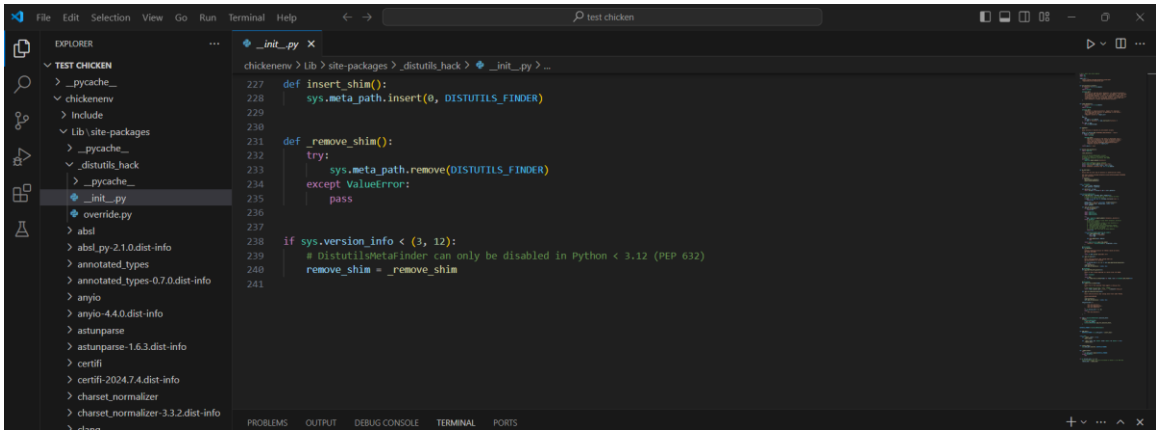


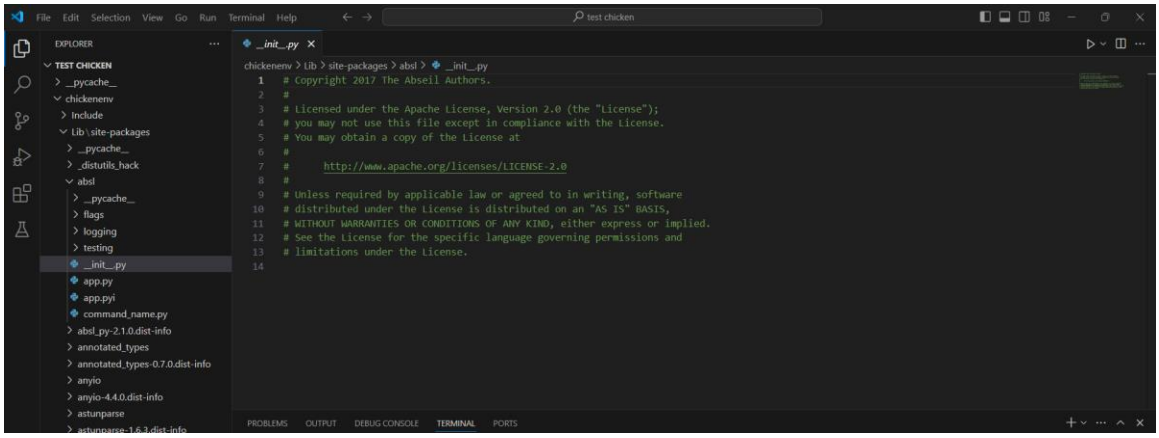


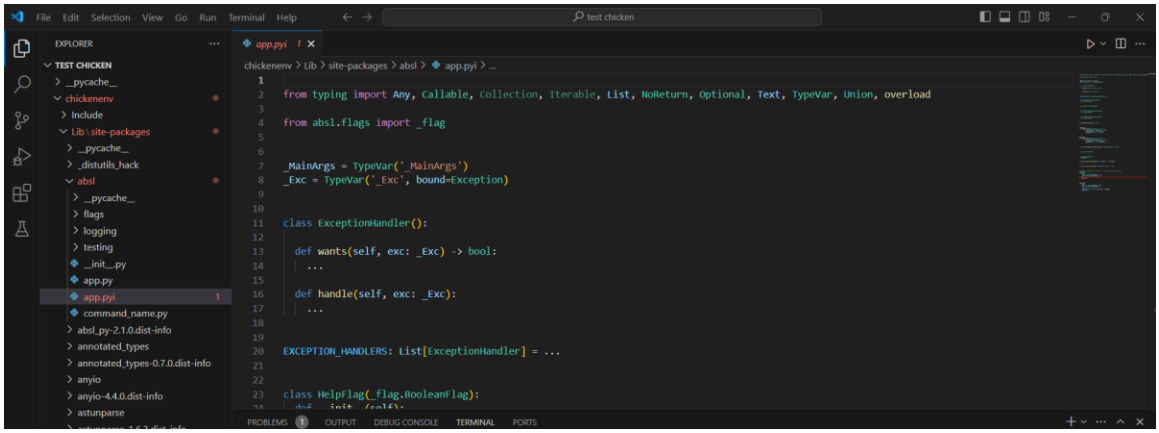


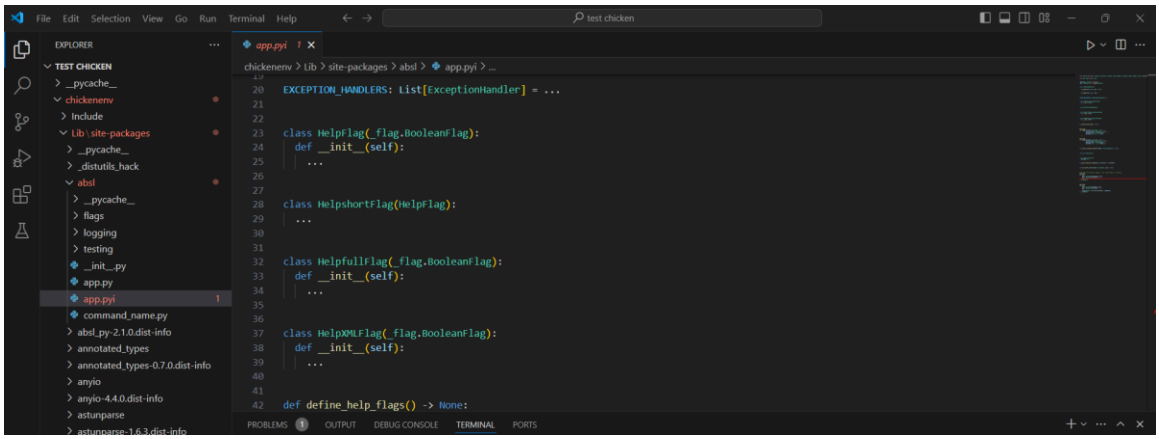


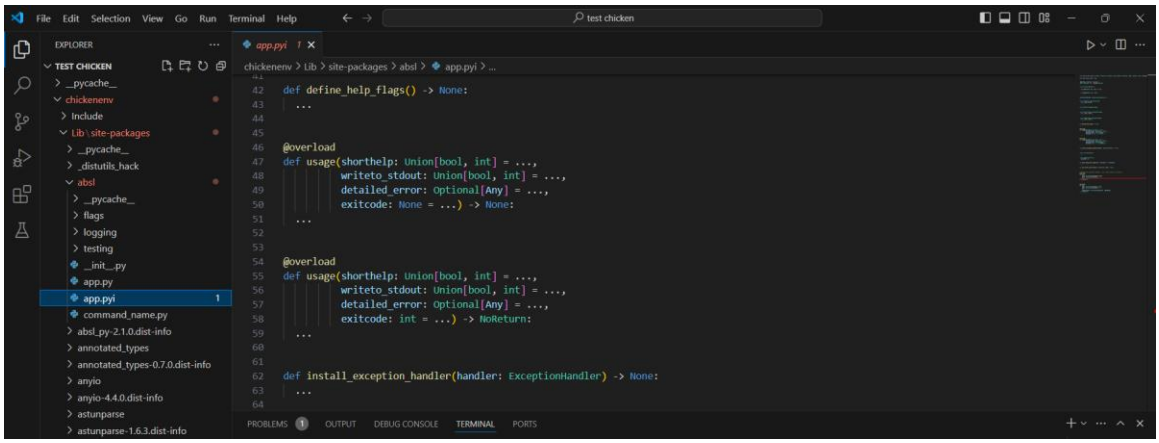


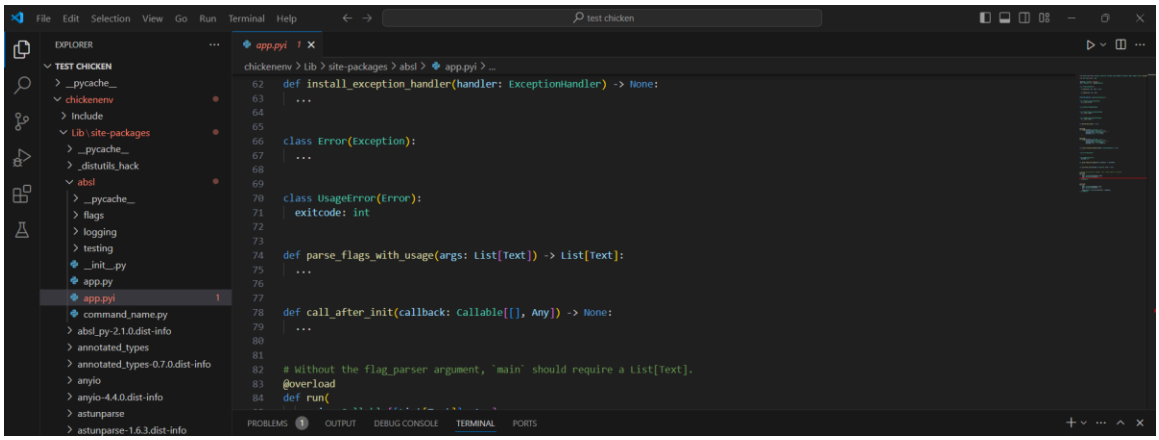


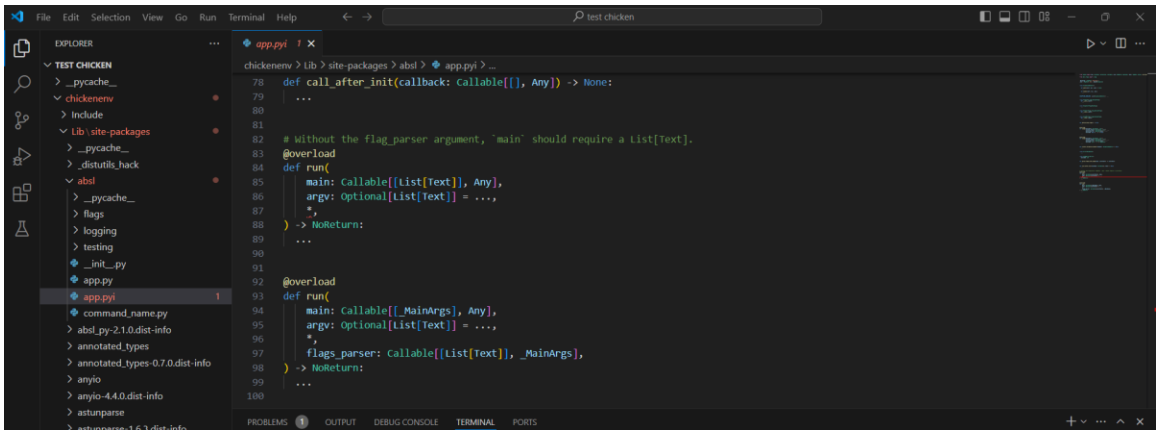


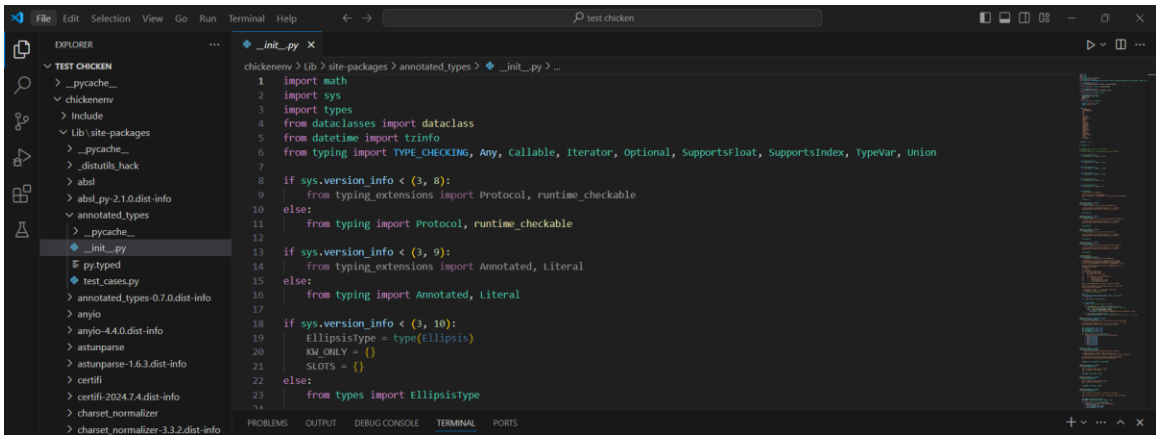


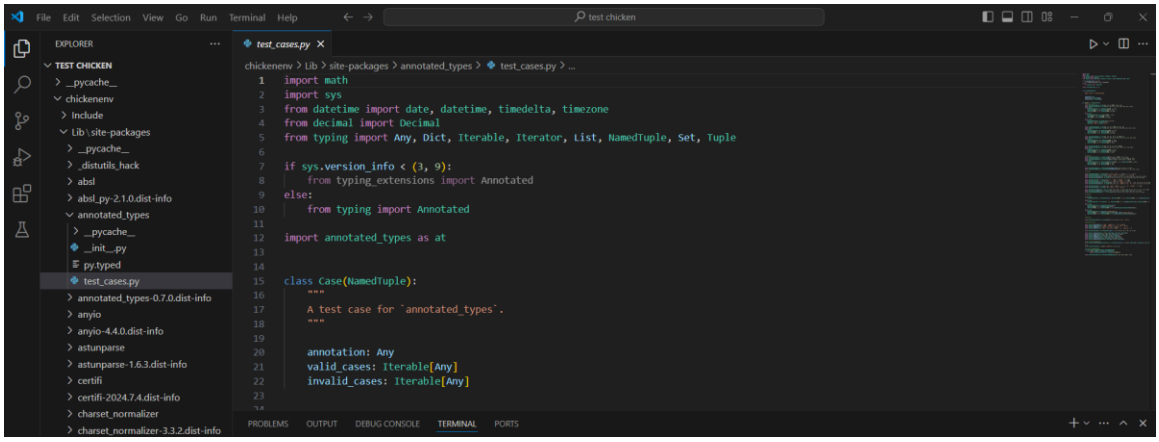


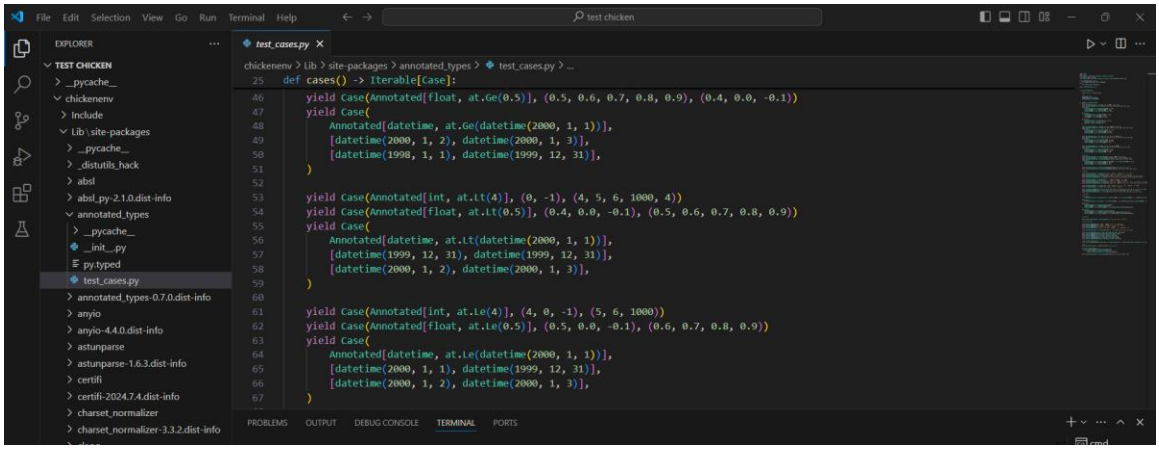


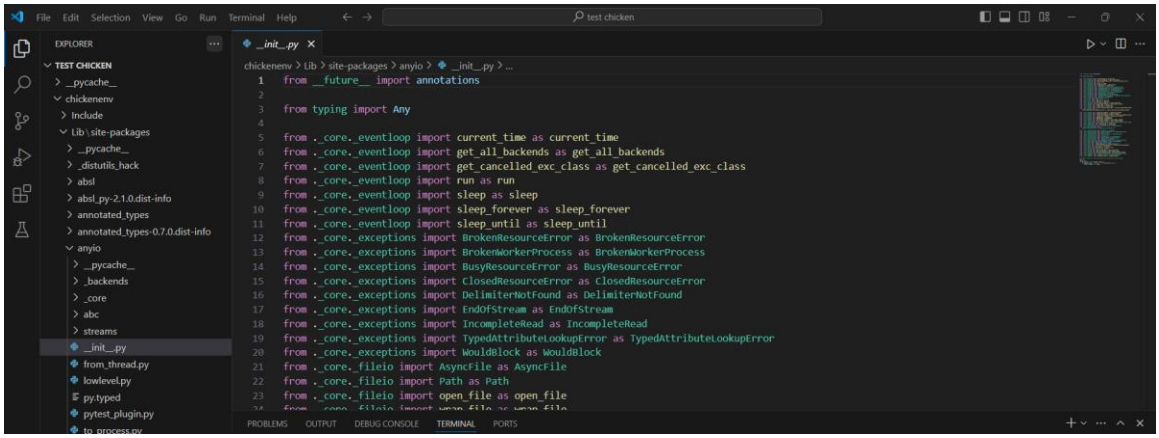


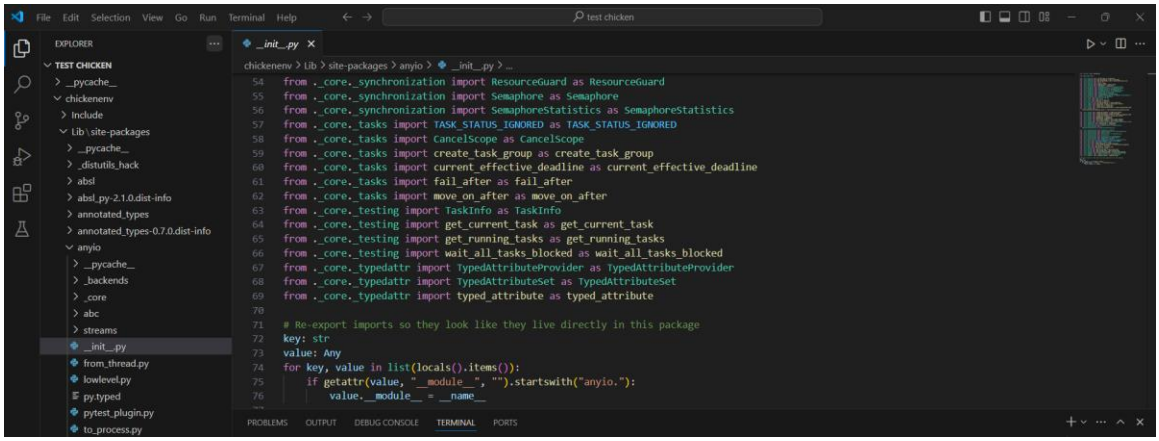


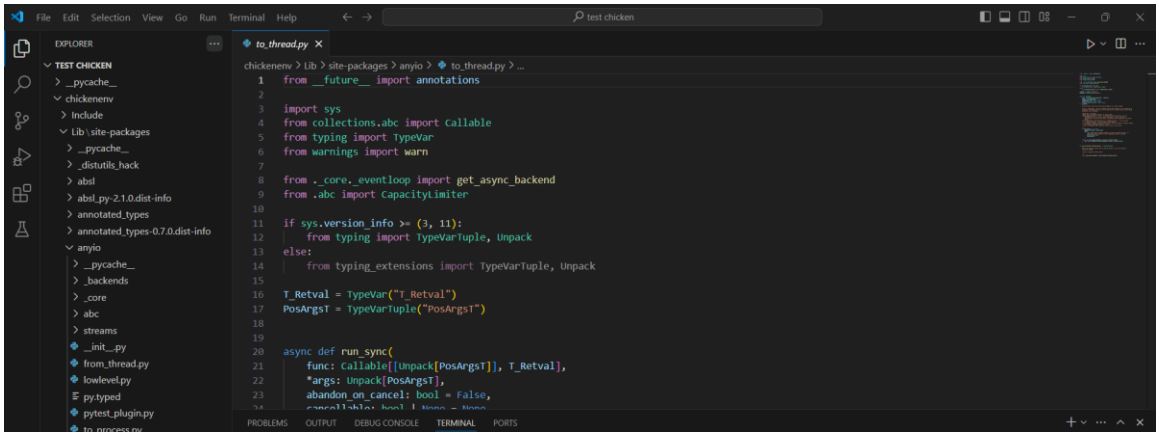


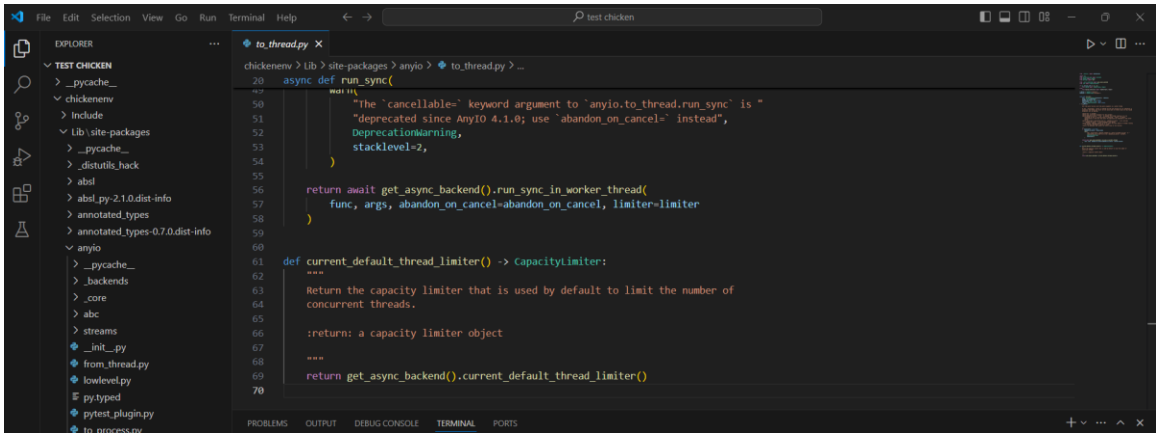


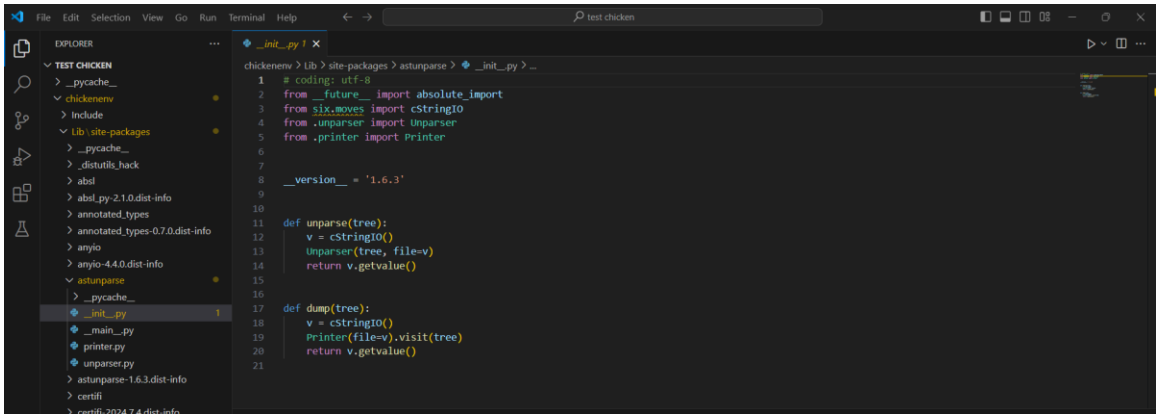


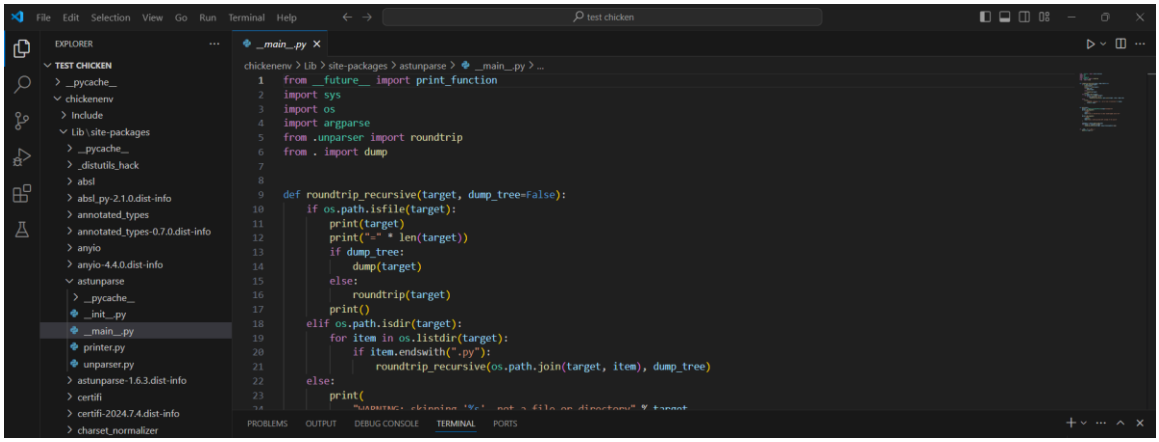


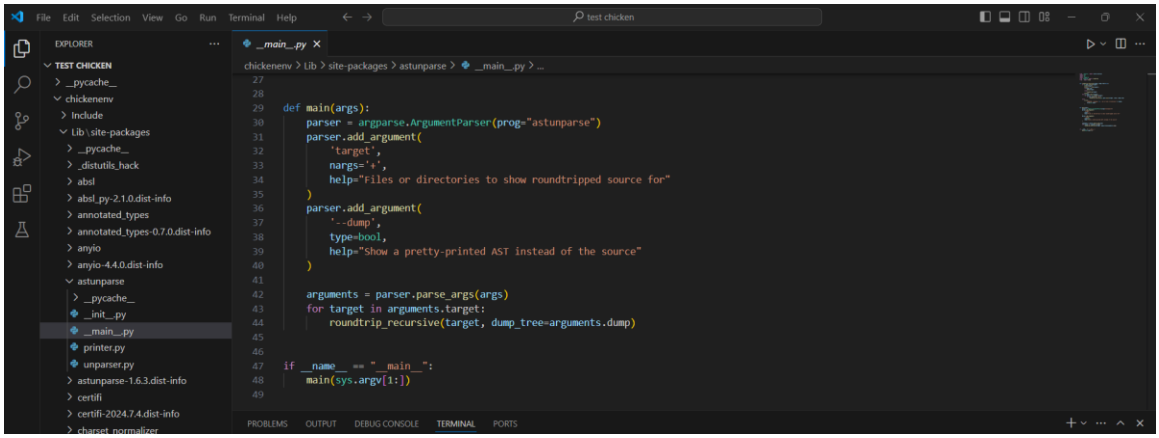


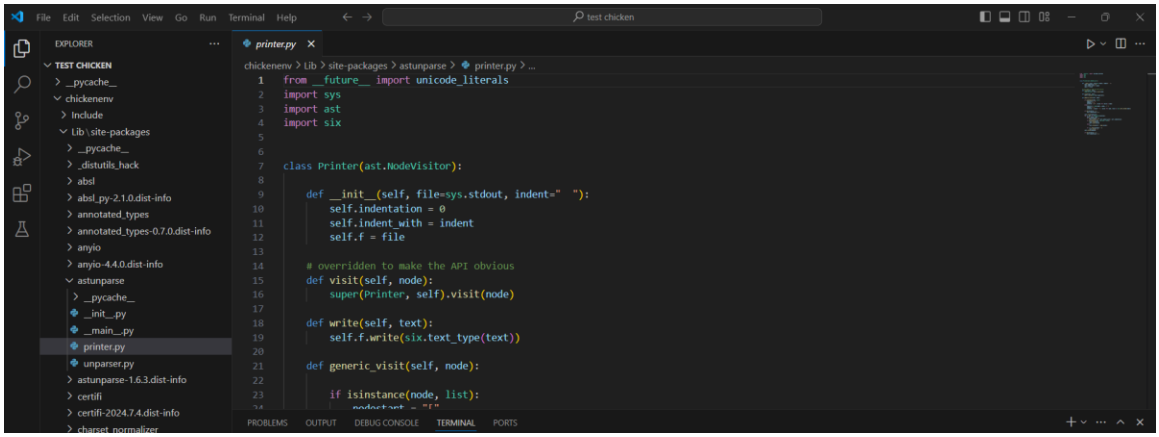


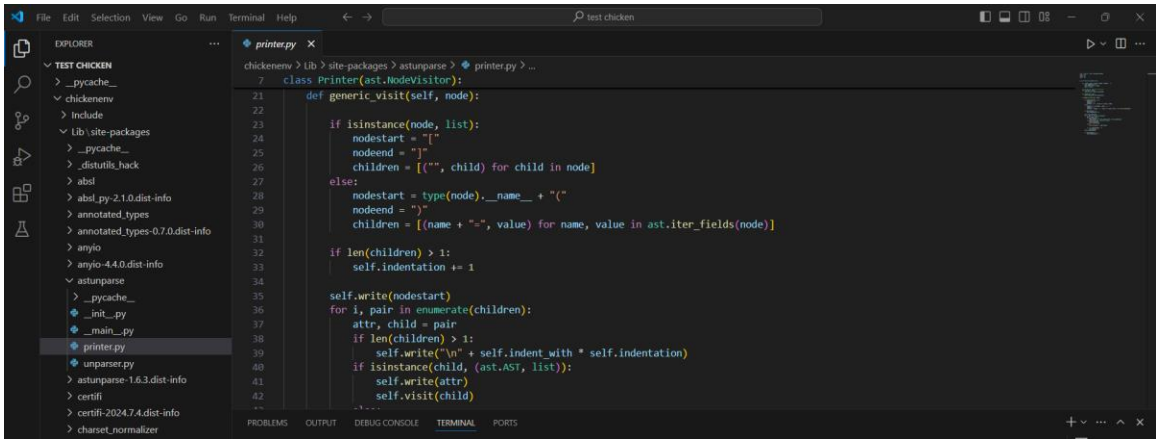


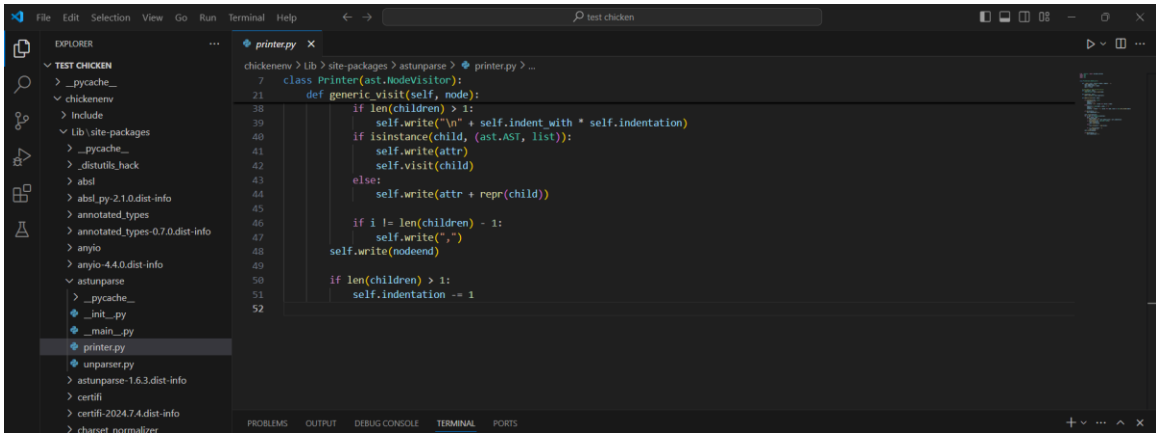


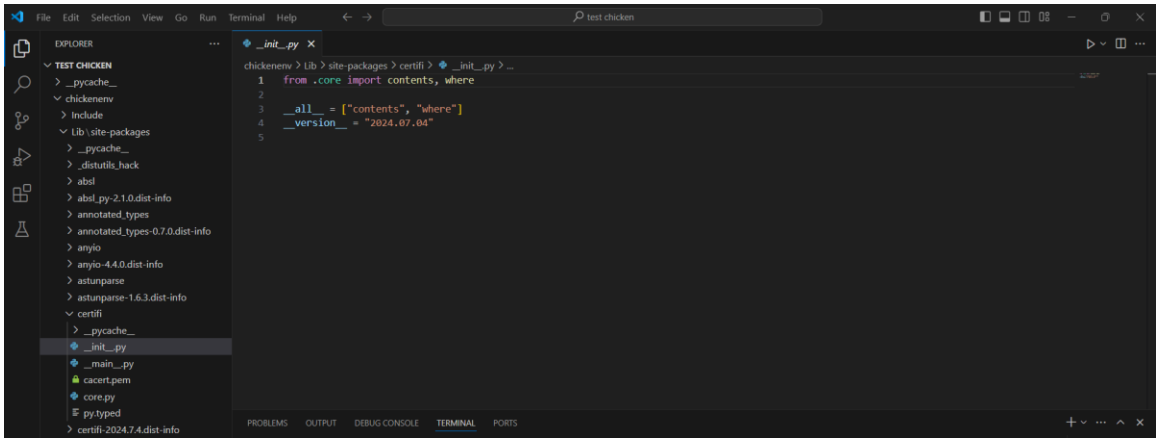


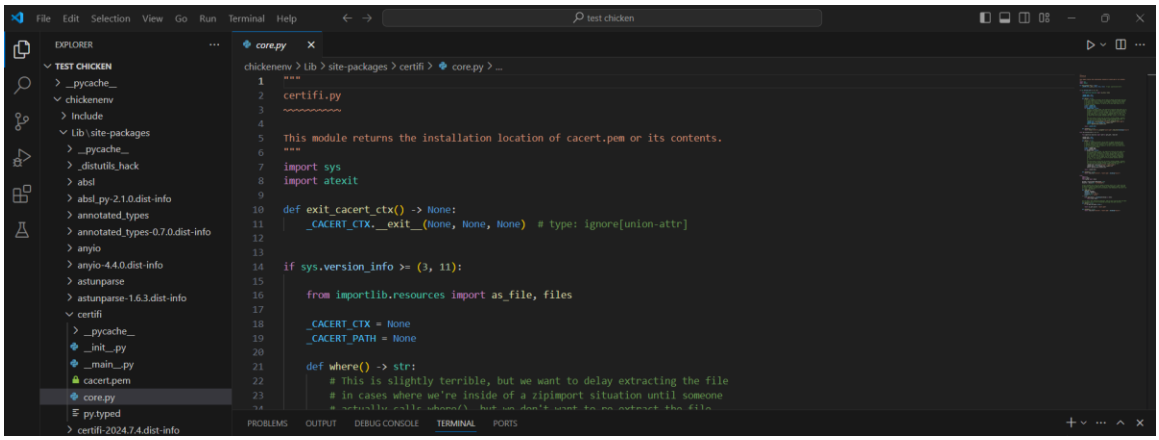


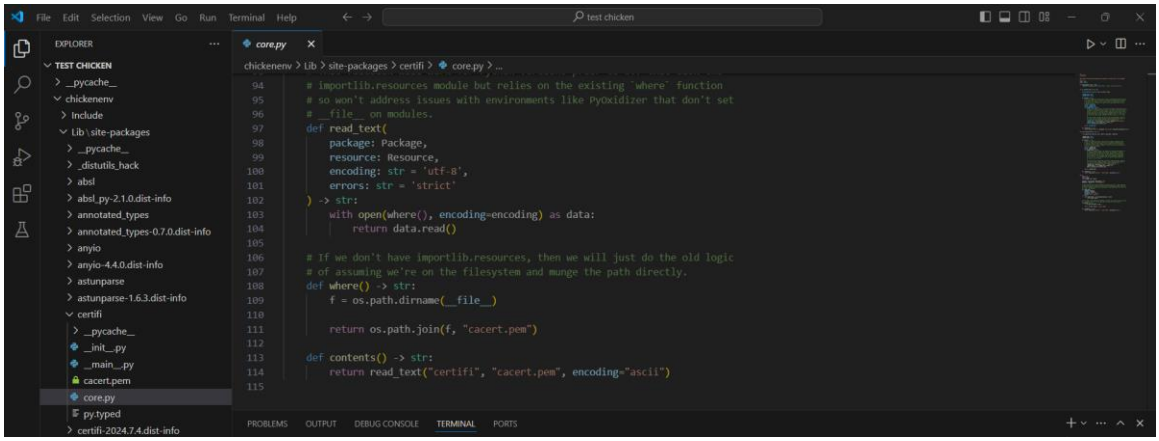


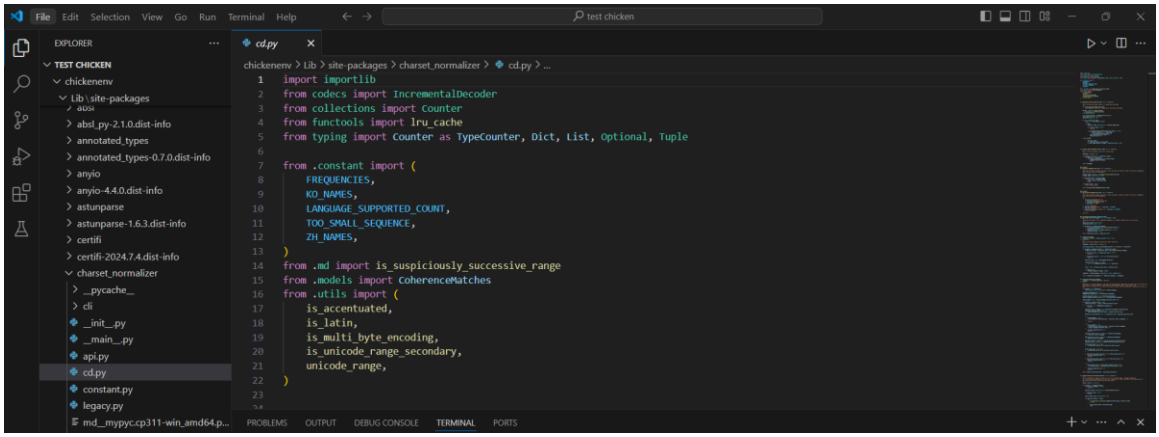


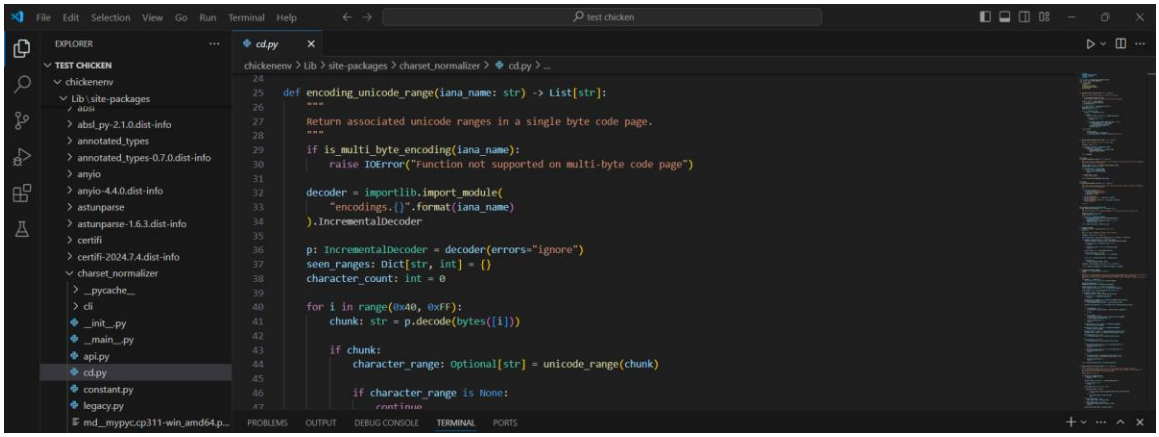


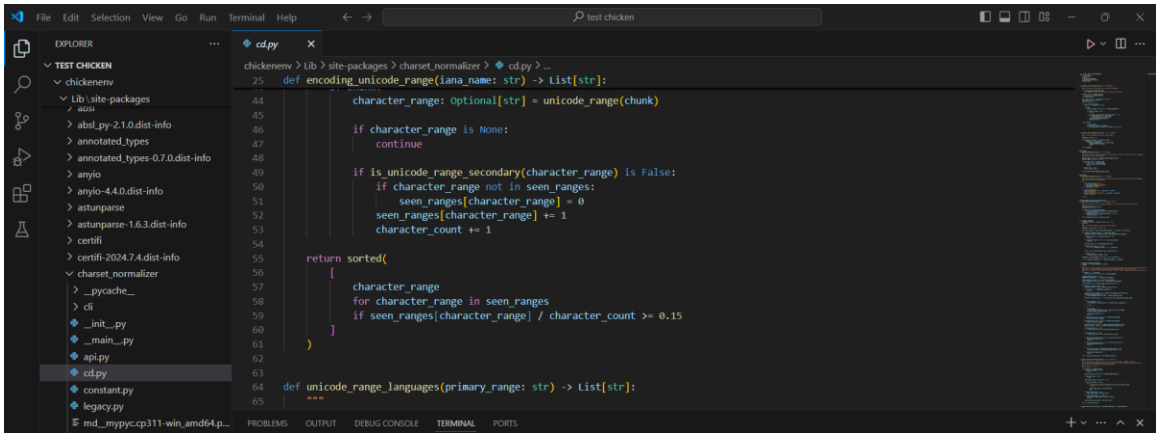


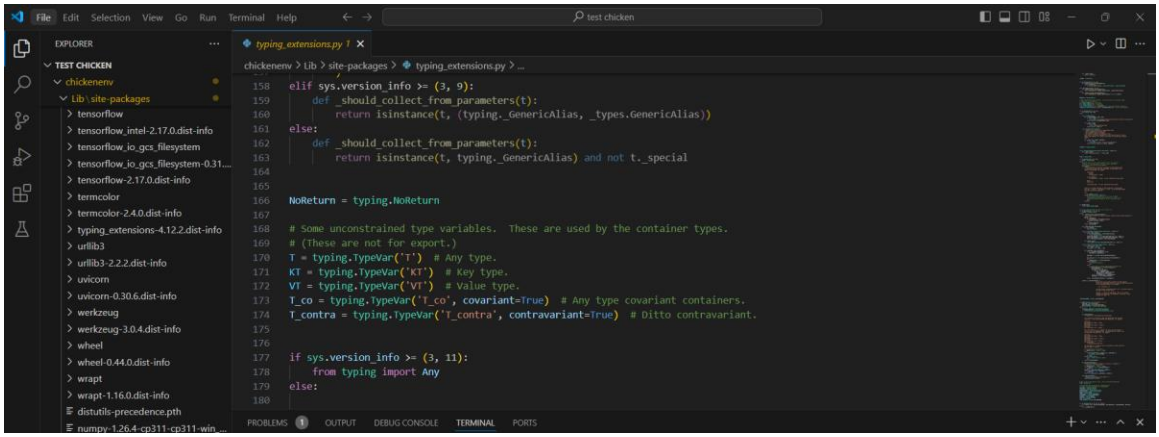


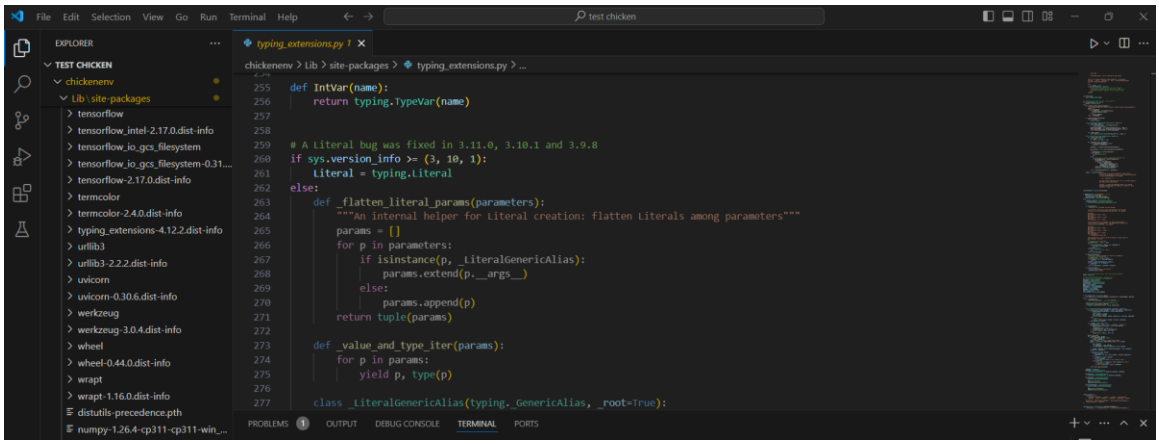


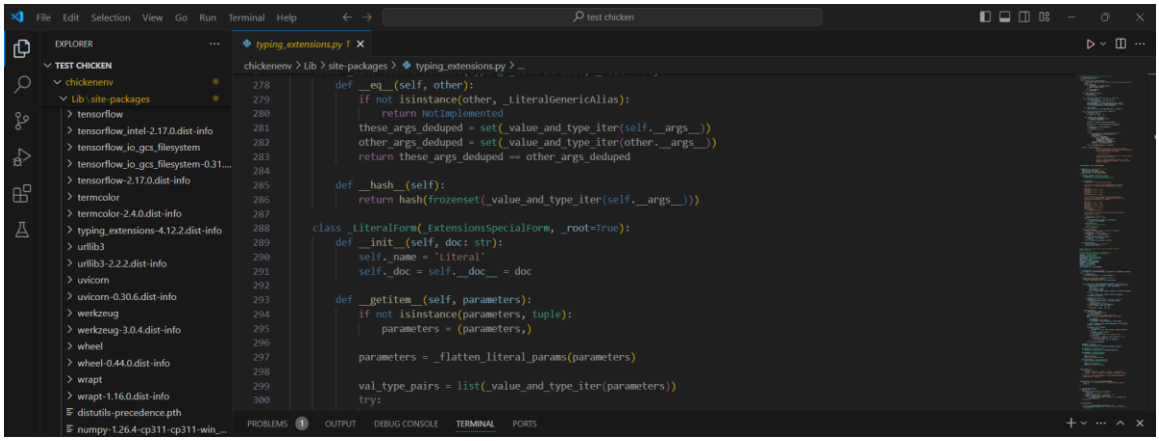


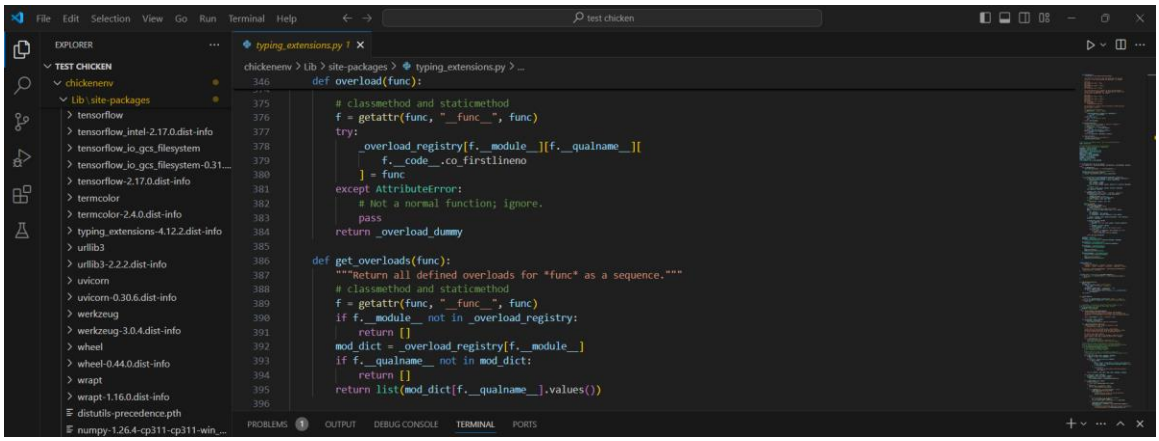


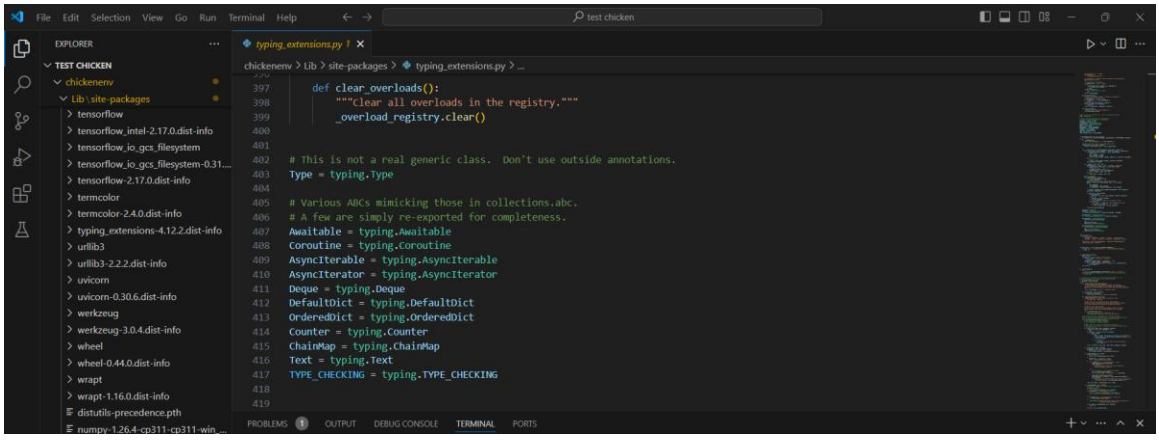


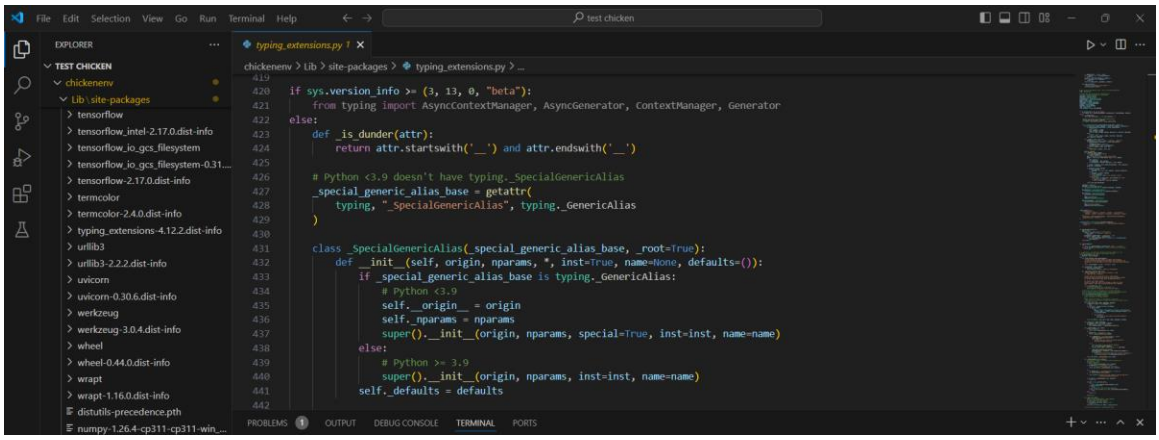


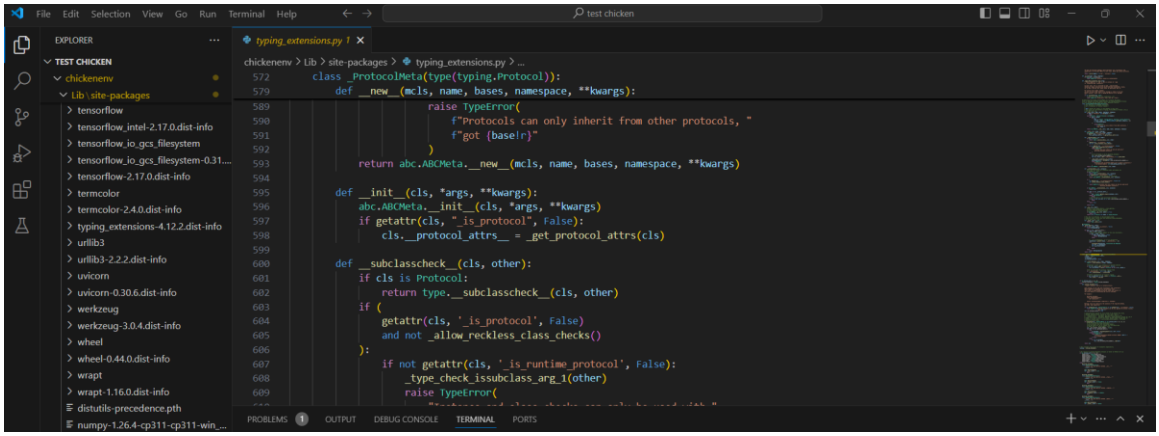


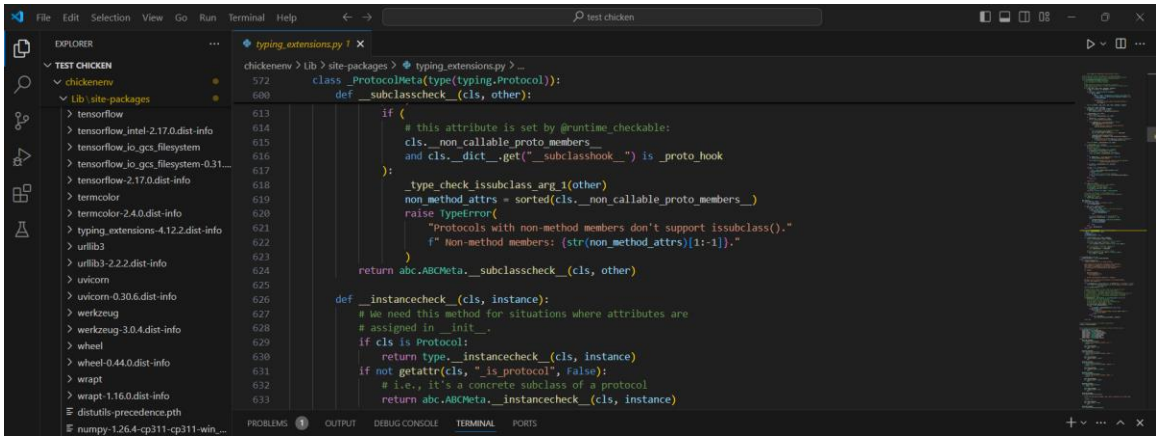


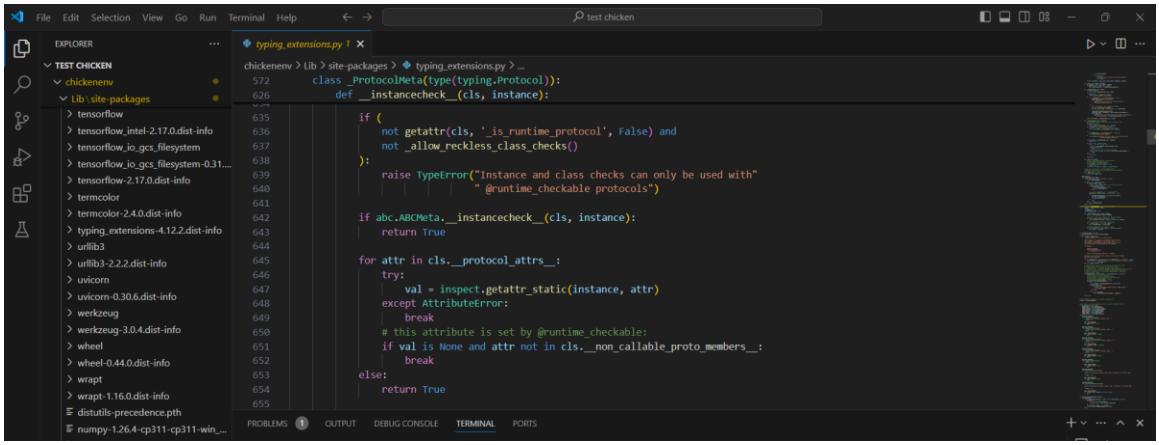


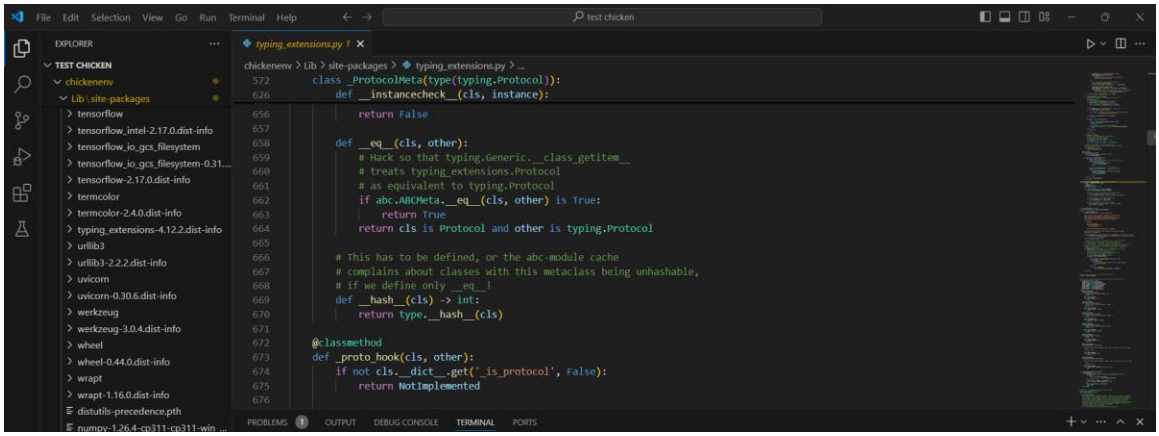


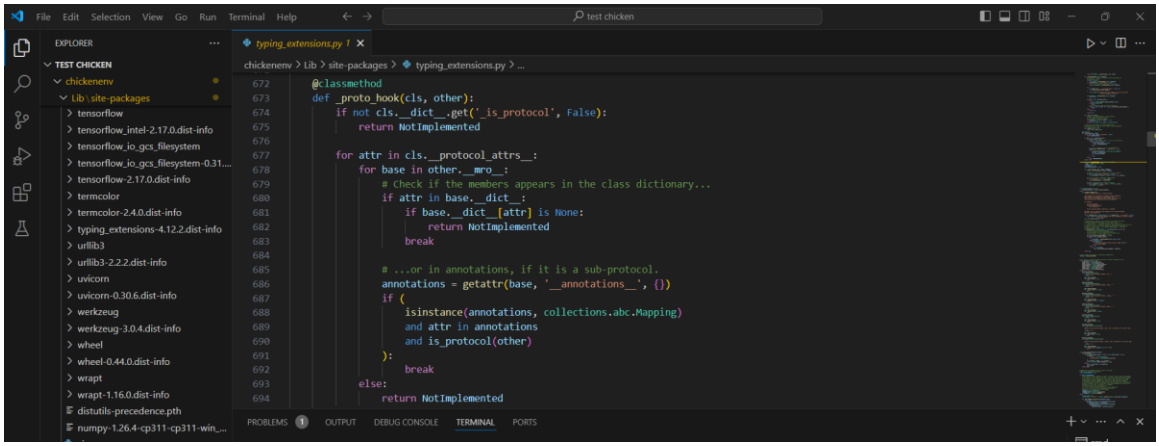


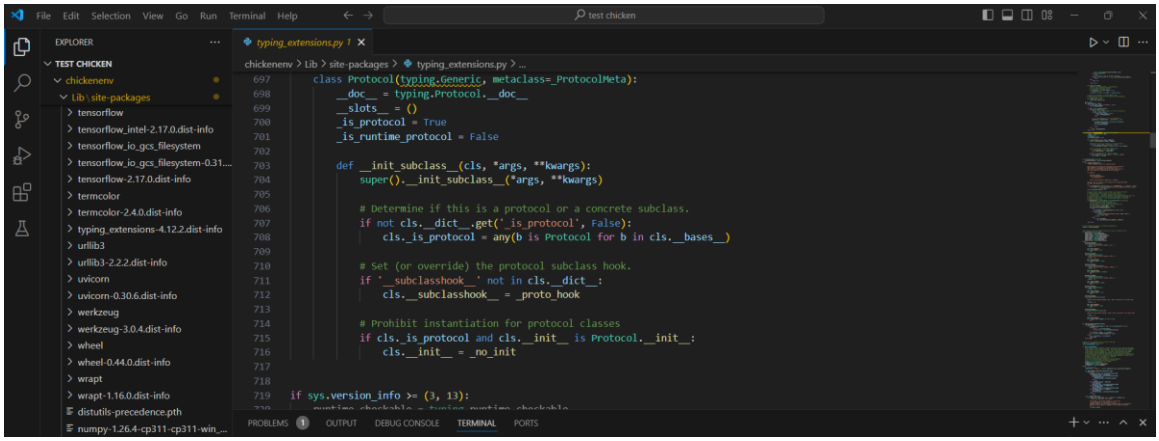


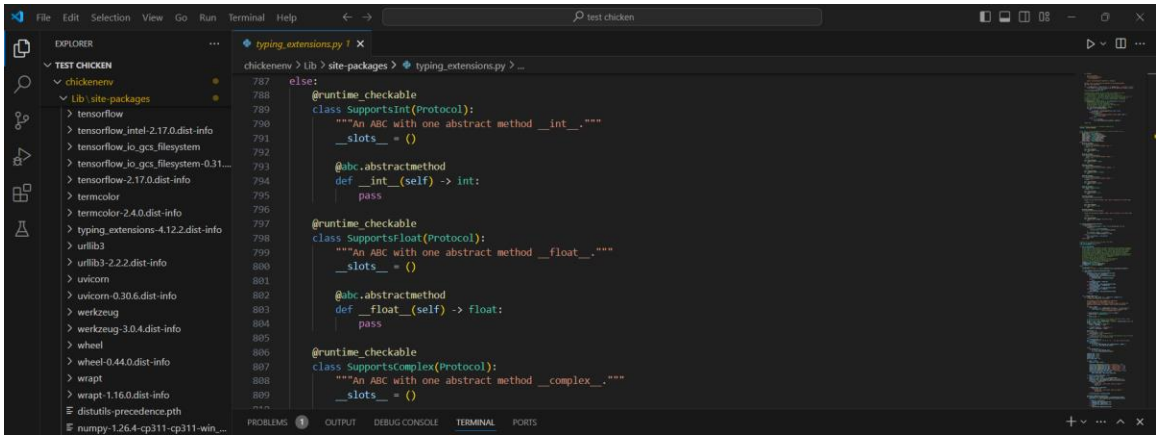


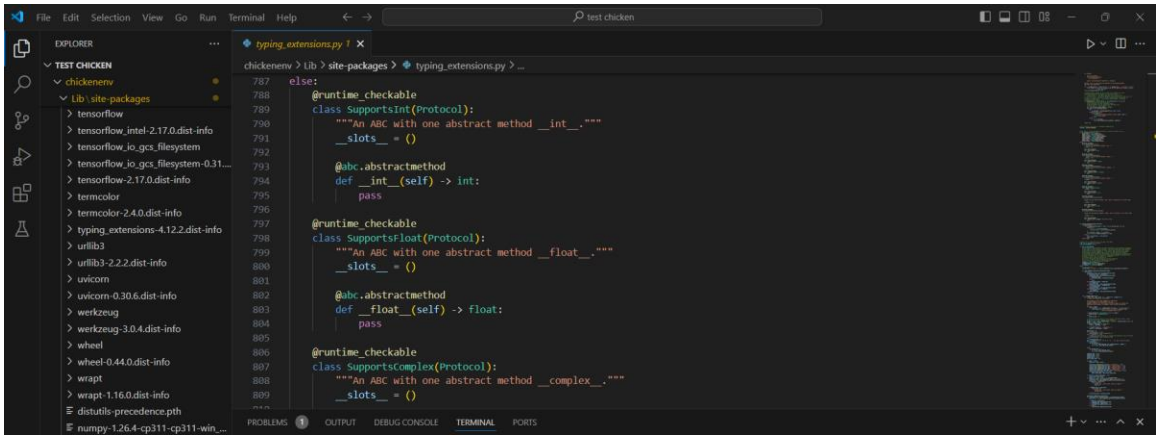


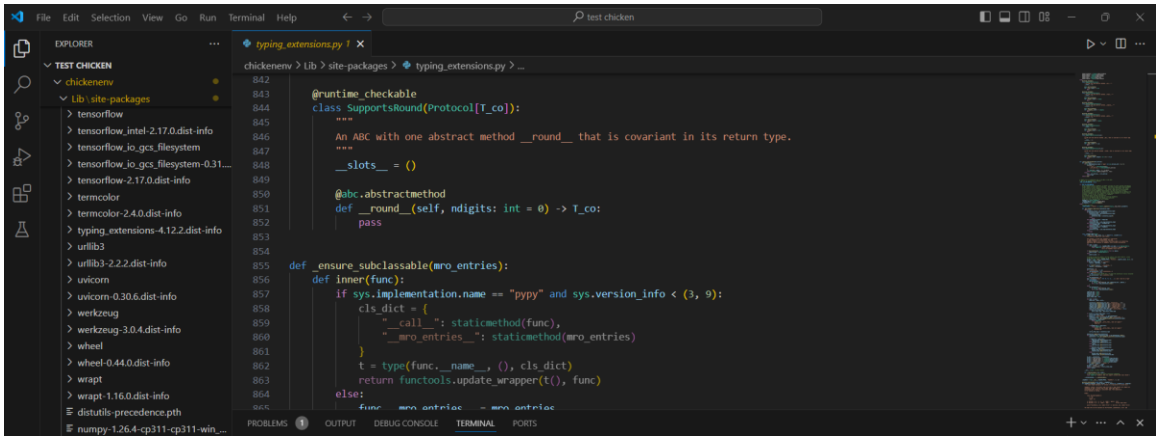




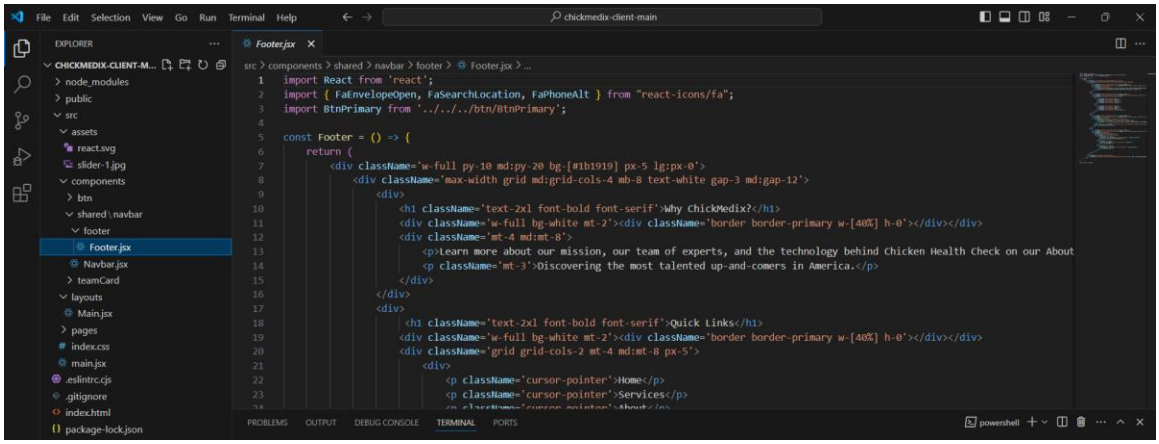




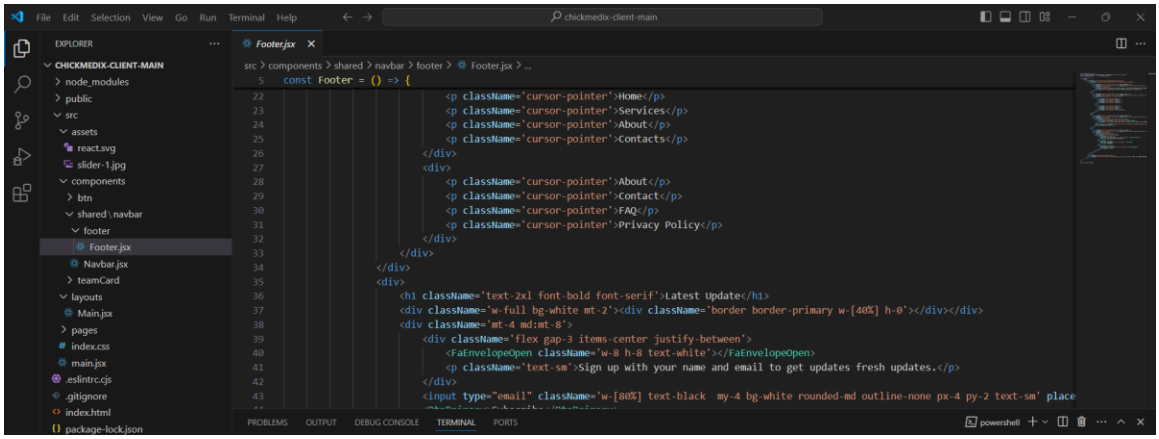




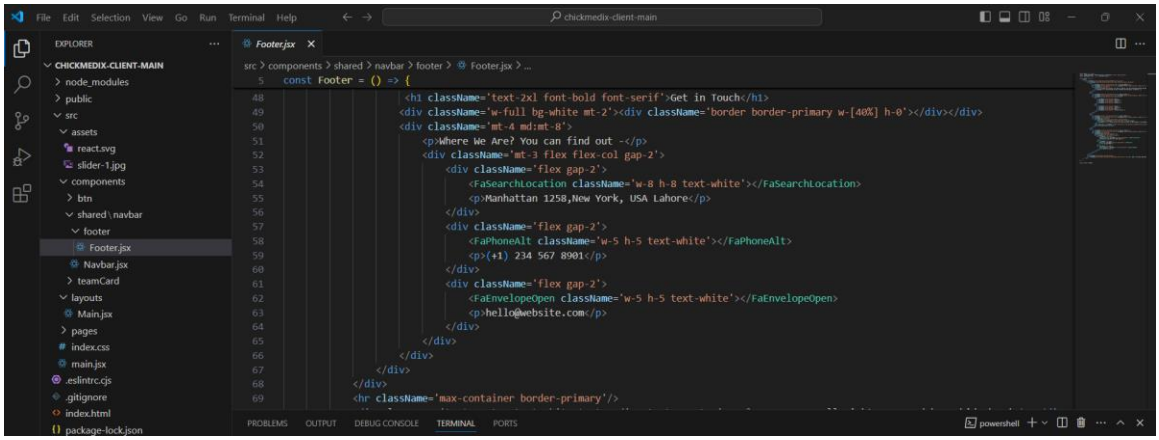
```
@runtime_checkable
class SupportsRound(Protocol[T_co]):
    """
    An ABC with one abstract method __round__ that is covariant in its return type.
    """
    __slots__ = ()

    @abc.abstractmethod
    def __round__(self, ndigits: int = 0) -> T_co:
        pass


def _ensure_subclassable(mro_entries):
    def inner(func):
        if sys.implementation.name == "pypy" and sys.version_info < (3, 9):
            cls_dict = {
                "__call__": staticmethod(func),
                "__mro_entries__": staticmethod(mro_entries)
            }
            t = type(func.__name__, (), cls_dict)
            return functools.update_wrapper(t(), func)
        else:
```



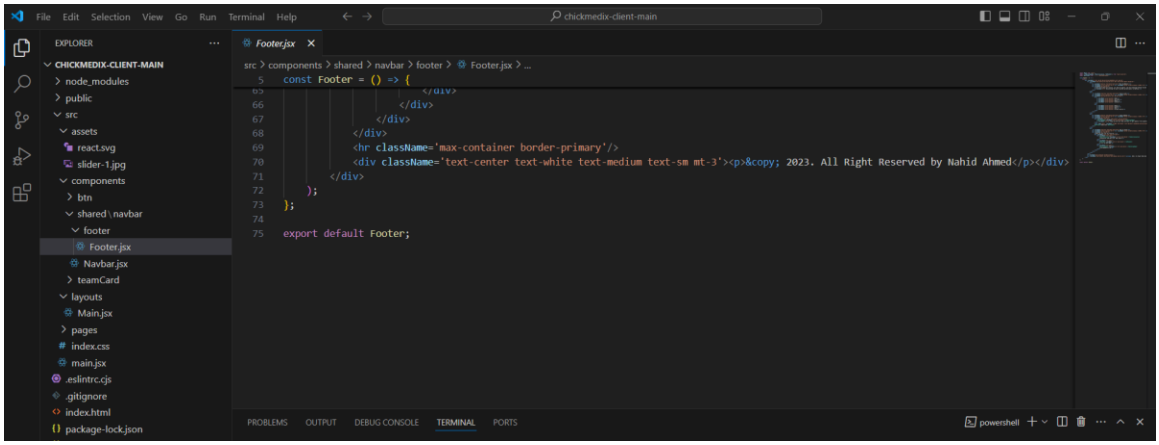
```
import React from 'react';
import { FaEnvelopeOpen, FaSearchLocation, FaPhoneAlt } from "react-icons/fa";
import BtnPrimary from '../../../btn/BtnPrimary';

const Footer = () => {
    return (
        <div className='w-full py-10 md:py-20 bg-[#1b1919] px-5 lg:px-0'>
            <div className='max-width grid md:grid-cols-4 mb-8 text-white gap-3 md:gap-12'>
                <div>
                    <h1 className='text-2xl font-bold font-serif'>Why ChickMedix?</h1>
                    <div className='w-full bg-white mt-2'><div className='border border-primary w-[40%] h-0'></div></div>
                    <div className='mt-4 md:mt-8'>
                        <p>Learn more about our mission, our team of experts, and the technology behind Chicken Health Check on our About
                        <p className='mt-3'>Discovering the most talented up-and-comers in America.</p>
                    </div>
                </div>
                <div>
                    <h1 className='text-2xl font-bold font-serif'>Quick Links</h1>
                    <div className='w-full bg-white mt-2'><div className='border border-primary w-[40%] h-0'></div></div>
                    <div className='grid grid-cols-2 mt-4 md:mt-8 px-5'>
                        <div>
                            <p className='cursor-pointer'>Home</p>
                            <p className='cursor-pointer'>Services</p>
```



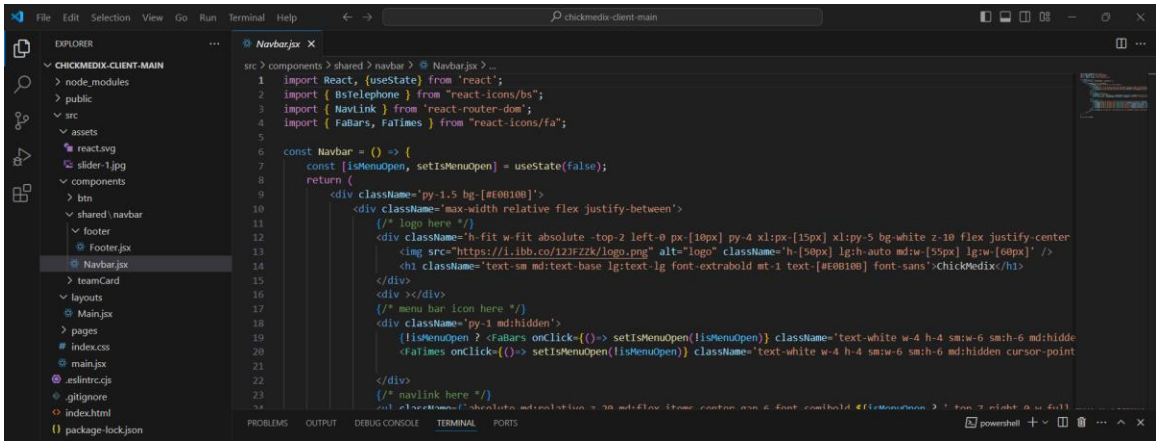
```
                            <p className='cursor-pointer'>Home</p>
                            <p className='cursor-pointer'>Services</p>
                            <p className='cursor-pointer'>About</p>
                            <p className='cursor-pointer'>Contacts</p>
                        </div>
                        <div>
                            <p className='cursor-pointer'>About</p>
                            <p className='cursor-pointer'>Contact</p>
                            <p className='cursor-pointer'>FAQ</p>
                            <p className='cursor-pointer'>Privacy Policy</p>
                        </div>
                    </div>
                </div>
                <div>
                    <h1 className='text-2xl font-bold font-serif'>Latest Update</h1>
                    <div className='w-full bg-white mt-2'><div className='border border-primary w-[40%] h-0'></div></div>
                    <div className='mt-4 md:mt-8'>
                        <div className='flex gap-3 items-center justify-between'>
                            <FaEnvelopeOpen className='w-8 h-8 text-white'></FaEnvelopeOpen>
                            <p className='text-sm'>Sign up with your name and email to get updates fresh updates.</p>
                        </div>
                        <input type="email" className='w-[80%] text-black  my-4 bg-white rounded-md outline-none px-4 py-2 text-sm' place
```



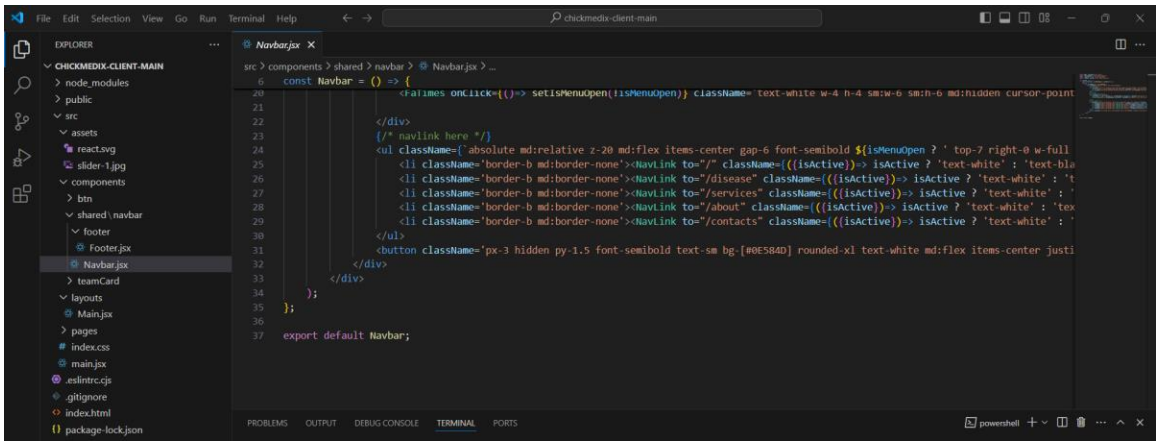
```
const Footer = () => {
                    <h1 className='text-2xl font-bold font-serif'>Get in Touch</h1>
                    <div className='w-full bg-white mt-2'><div className='border border-primary w-[40%] h-0'></div></div>
                    <div className='mt-4 md:mt-8'>
                        <p>Where We Are? You can find out -</p>
                        <div className='mt-3 flex flex-col gap-2'>
                            <div className='flex gap-2'>
                                <FaSearchLocation className='w-8 h-8 text-white'></FaSearchLocation>
                                <p>Manhattan 1258,New York, USA Lahore</p>
                            </div>
                            <div className='flex gap-2'>
                                <FaPhoneAlt className='w-5 h-5 text-white'></FaPhoneAlt>
                                <p>(+1) 234 567 8901</p>
                            </div>
                            <div className='flex gap-2'>
                                <FaEnvelopeOpen className='w-5 h-5 text-white'></FaEnvelopeOpen>
                                <p>hello@website.com</p>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            <hr className='max-container border-primary'/>
```



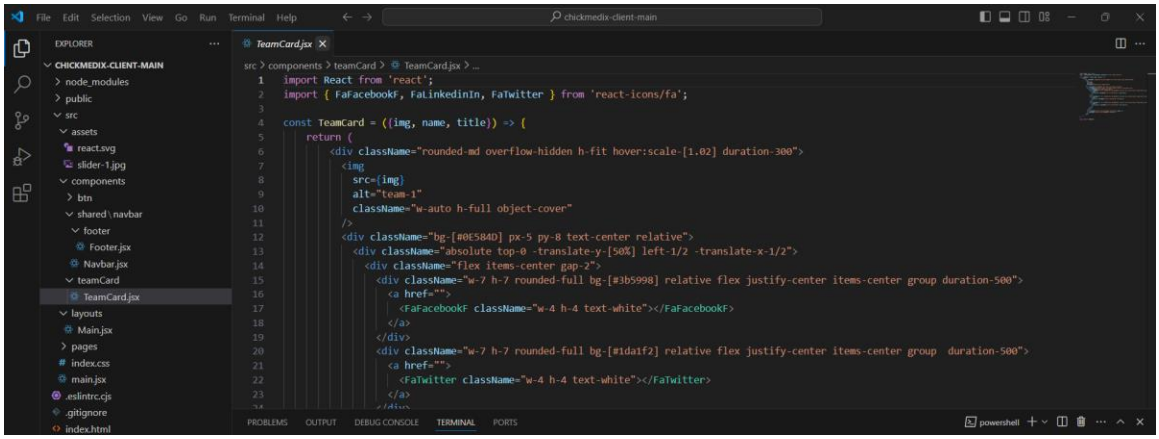
```
const Footer = () => {
                </div>
            </div>
        </div>
        <hr className='max-container border-primary'/>
        <div className='text-center text-white text-medium text-sm mt-3'><p>&copy; 2023. All Right Reserved by Nahid Ahmed</p></div>
    </div>
  );
};

export default Footer;
```



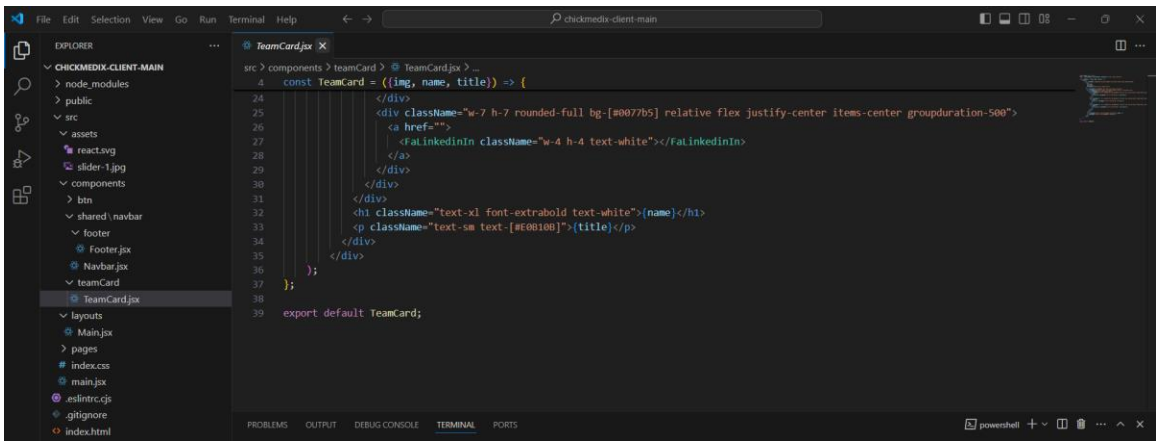
```
import React, {useState} from 'react';
import { BsTelephone } from "react-icons/bs";
import { NavLink } from 'react-router-dom';
import { FaBars, FaTimes } from "react-icons/fa";

const Navbar = () => {
    const [isMenuOpen, setIsMenuOpen] = useState(false);
    return (
        <div className='py-1.5 bg-[#E0B10B]'>
            <div className='max-width relative flex justify-between'>
                {/* logo here */}
                <div className='h-fit w-fit absolute -top-2 left-0 px-[10px] py-4 xl:px-[15px] xl:py-5 bg-white z-10 flex justify-center
                    <img src="https://i.ibb.co/12JFZZk/logo.png" alt="logo" className='h-[50px] lg:h-auto md:w-[55px] lg:w-[60px]' />
                    <h1 className='text-sm md:text-base lg:text-lg font-extrabold mt-1 text-[#E0B10B] font-sans'>ChickMedix</h1>
                </div>
                <div ></div>
                {/* menu bar icon here */}
                <div className='py-1 md:hidden'>
                    {!isMenuOpen ? <FaBars onClick={()=> setIsMenuOpen(!isMenuOpen)} className='text-white w-4 h-4 sm:w-6 sm:h-6 md:hidde
                    <FaTimes onClick={()=> setIsMenuOpen(!isMenuOpen)} className='text-white w-4 h-4 sm:w-6 sm:h-6 md:hidden cursor-point

                </div>
                {/* navlink here */}
```



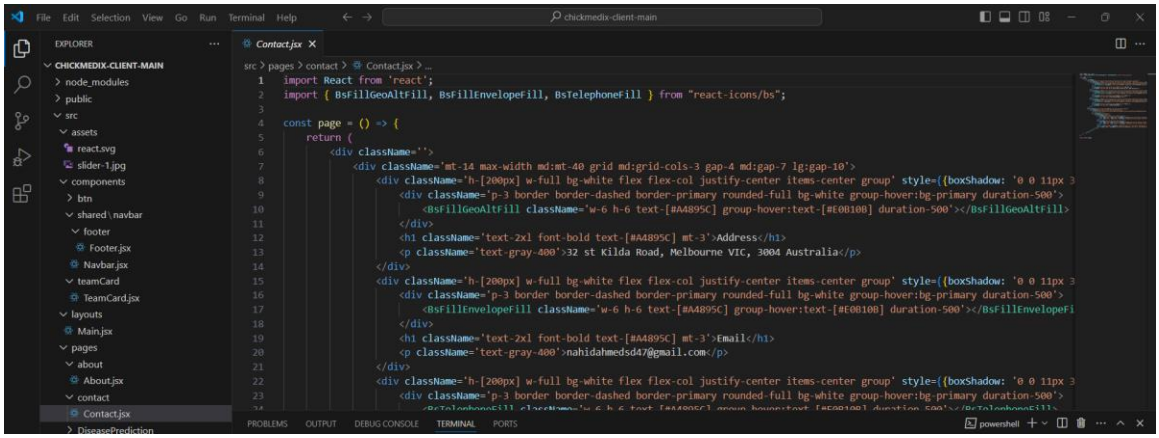
```jsx
const Navbar = () => {
                    <FaTimes onClick={()=> setIsMenuOpen(!isMenuOpen)} className="text-white w-4 h-4 sm:w-6 sm:h-6 md:hidden cursor-point
                </div>
                {/* navlink here */}
                <ul className={`absolute md:relative z-20 md:flex items-center gap-6 font-semibold ${isMenuOpen ? ' top-7 right-0 w-full
                  <li className='border-b md:border-none'><NavLink to="/" className={({isActive})=> isActive ? 'text-white' : 'text-bla
                  <li className='border-b md:border-none'><NavLink to="/disease" className={({isActive})=> isActive ? 'text-white' : 't
                  <li className='border-b md:border-none'><NavLink to="/services" className={({isActive})=> isActive ? 'text-white' : '
                  <li className='border-b md:border-none'><NavLink to="/about" className={({isActive})=> isActive ? 'text-white' : 'tex
                  <li className='border-b md:border-none'><NavLink to="/contacts" className={({isActive})=> isActive ? 'text-white' : '
                </ul>
                <button className='px-3 hidden py-1.5 font-semibold text-sm bg-[#0E584D] rounded-xl text-white md:flex items-center justi
              </div>
            </div>
        );
    };

    export default Navbar;
```



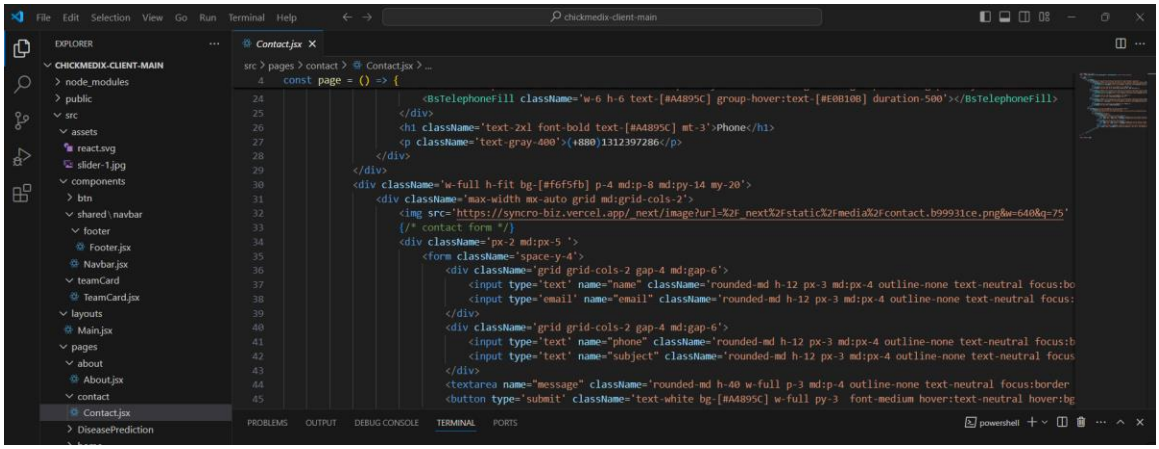
```jsx
import React from 'react';
import { FaFacebookF, FaLinkedinIn, FaTwitter } from 'react-icons/fa';

const TeamCard = ({img, name, title}) => {
    return (
        <div className="rounded-md overflow-hidden h-fit hover:scale-[1.02] duration-300">
          <img
            src={img}
            alt="team-1"
            className="w-auto h-full object-cover"
          />
          <div className="bg-[#0E584D] px-5 py-8 text-center relative">
            <div className="absolute top-0 -translate-y-[50%] left-1/2 -translate-x-1/2">
              <div className="flex items-center gap-2">
                <div className="w-7 h-7 rounded-full bg-[#3b5998] relative flex justify-center items-center group duration-500">
                  <a href="">
                    <FaFacebookF className="w-4 h-4 text-white"></FaFacebookF>
                  </a>
                </div>
                <div className="w-7 h-7 rounded-full bg-[#1da1f2] relative flex justify-center items-center group  duration-500">
                  <a href="">
                    <FaTwitter className="w-4 h-4 text-white"></FaTwitter>
                  </a>
```



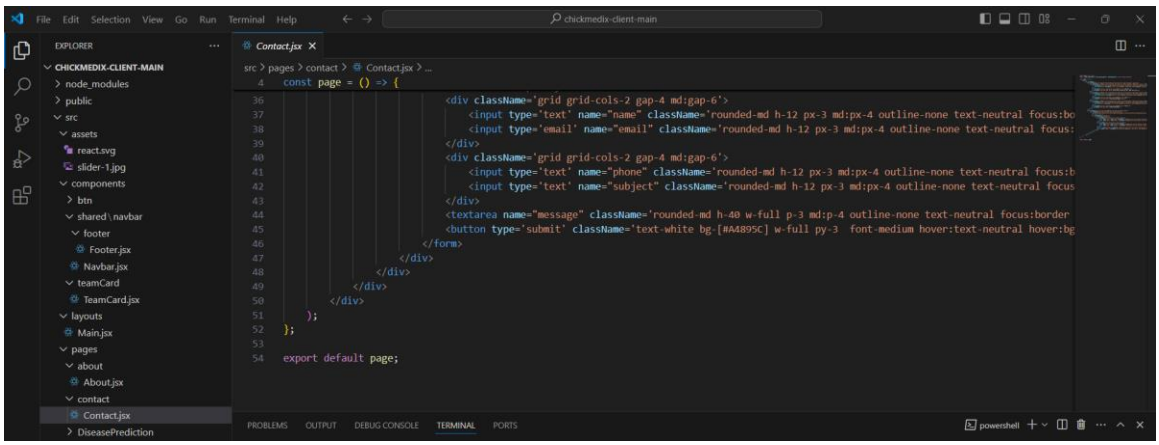
```jsx
const TeamCard = ({img, name, title}) => {
                </div>
                <div className="w-7 h-7 rounded-full bg-[#0077b5] relative flex justify-center items-center groupduration-500">
                  <a href="">
                    <FaLinkedinIn className="w-4 h-4 text-white"></FaLinkedinIn>
                  </a>
                </div>
              </div>
            </div>
            <h1 className="text-xl font-extrabold text-white">{name}</h1>
            <p className="text-sm text-[#E0B10B]">{title}</p>
          </div>
        </div>
    );
};

export default TeamCard;
```



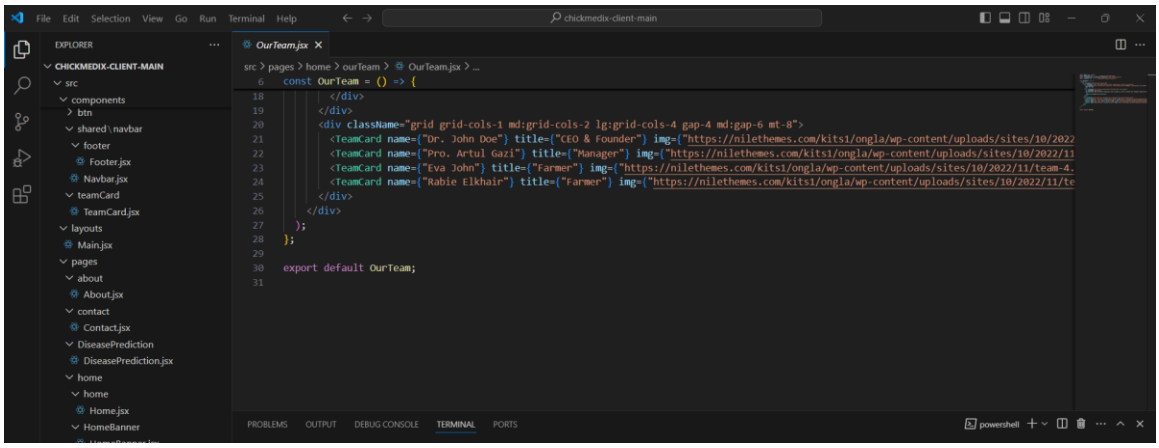
```
import React from 'react';
import { BsFillGeoAltFill, BsFillEnvelopeFill, BsTelephoneFill } from "react-icons/bs";

const page = () => {
    return (
        <div className=''>
            <div className='mt-14 max-width md:mt-40 grid md:grid-cols-3 gap-4 md:gap-7 lg:gap-10'>
                <div className='h-[200px] w-full bg-white flex flex-col justify-center items-center group' style={{boxShadow: '0 0 11px 3
                    <div className='p-3 border border-dashed border-primary rounded-full bg-white group-hover:bg-primary duration-500'>
                        <BsFillGeoAltFill className='w-6 h-6 text-[#A4895C] group-hover:text-[#E0B10B] duration-500'></BsFillGeoAltFill>
                    </div>
                    <h1 className='text-2xl font-bold text-[#A4895C] mt-3'>Address</h1>
                    <p className='text-gray-400'>32 st Kilda Road, Melbourne VIC, 3004 Australia</p>
                </div>
                <div className='h-[200px] w-full bg-white flex flex-col justify-center items-center group' style={{boxShadow: '0 0 11px 3
                    <div className='p-3 border border-dashed border-primary rounded-full bg-white group-hover:bg-primary duration-500'>
                        <BsFillEnvelopeFill className='w-6 h-6 text-[#A4895C] group-hover:text-[#E0B10B] duration-500'></BsFillEnvelopeFi
                    </div>
                    <h1 className='text-2xl font-bold text-[#A4895C] mt-3'>Email</h1>
                    <p className='text-gray-400'>nahidahmedsd47@gmail.com</p>
                </div>
                <div className='h-[200px] w-full bg-white flex flex-col justify-center items-center group' style={{boxShadow: '0 0 11px 3
                    <div className='p-3 border border-dashed border-primary rounded-full bg-white group-hover:bg-primary duration-500'>
```

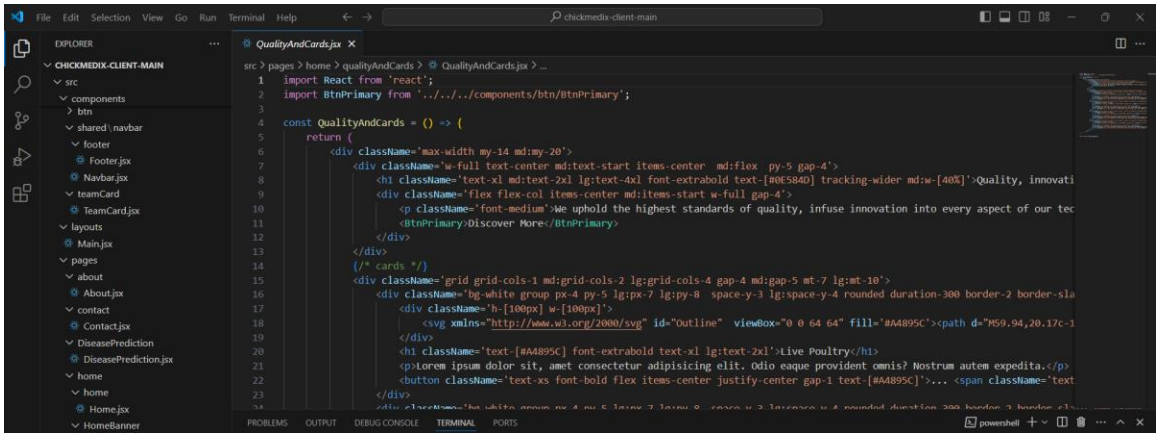


```
const page = () => {
                        <BsTelephoneFill className='w-6 h-6 text-[#A4895C] group-hover:text-[#E0B10B] duration-500'></BsTelephoneFill>
                    </div>
                    <h1 className='text-2xl font-bold text-[#A4895C] mt-3'>Phone</h1>
                    <p className='text-gray-400'>(+880)1312397286</p>
                </div>
            </div>
            <div className='w-full h-fit bg-[#f6f5fb] p-4 md:p-8 md:py-14 my-20'>
                <div className='max-width mx-auto grid md:grid-cols-2'>
                    <img src='https://syncro-biz.vercel.app/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fcontact.b9993ce.png&w=640&q=75'
                    {/* contact form */}
                    <div className='px-2 md:px-5 '>
                        <form className='space-y-4'>
                            <div className='grid grid-cols-2 gap-4 md:gap-6'>
                                <input type='text' name="name" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus:bo
                                <input type='email' name="email" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus:
                            </div>
                            <div className='grid grid-cols-2 gap-4 md:gap-6'>
                                <input type='text' name="phone" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus:b
                                <input type='text' name="subject" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus
                            </div>
                            <textarea name="message" className='rounded-md h-40 w-full p-3 md:p-4 outline-none text-neutral focus:border
                            <button type='submit' className='text-white bg-[#A4895C] w-full py-3  font-medium hover:text-neutral hover:bg
```



```
const page = () => {
                            <div className='grid grid-cols-2 gap-4 md:gap-6'>
                                <input type='text' name="name" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus:bo
                                <input type='email' name="email" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus:
                            </div>
                            <div className='grid grid-cols-2 gap-4 md:gap-6'>
                                <input type='text' name="phone" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus:b
                                <input type='text' name="subject" className='rounded-md h-12 px-3 md:px-4 outline-none text-neutral focus
                            </div>
                            <textarea name="message" className='rounded-md h-40 w-full p-3 md:p-4 outline-none text-neutral focus:border
                            <button type='submit' className='text-white bg-[#A4895C] w-full py-3  font-medium hover:text-neutral hover:bg
                        </form>
                    </div>
                </div>
            </div>
        </div>
    );
};

export default page;
```





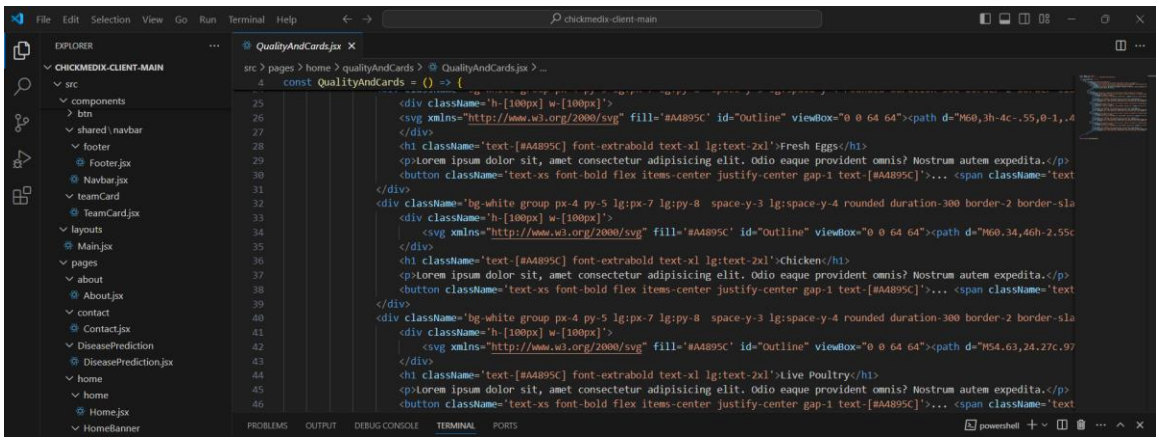


Figure 10: Code Sample

10.2 Probability problem break down

Dividing the problem into smaller manageable parts is crucial to tackling the probability issues related to an AI-powered online app for Chicken Disease Detection. In these kinds of systems, probability concerns are typically related to decision-making under uncertainty, model accuracy, and confidence in illness prediction. Through the decomposition of probability issues related to a web application for detecting chicken diseases, we may tackle significant obstacles such precision of the model, reliability rating, longevity, managing uncommon illnesses, and frequency of mistakes. To guarantee that the model continues to be useful and dependable in identifying chicken diseases, resolving these issues entails putting strong metrics for assessment, calibration methodologies, surveillance of performance, and adaption strategies into practice.

- **Data security and protection**: Ensuring that scanned content is handled properly to avoid unwanted access or exposure, particularly source code and files for configuration. applying robust encryption techniques, especially when transmitting scan findings or maintaining them in databases, to protect data while it's in transit and at rest.
- **Debugging and Testing**: Determining and resolving issues, such as back-end and front-end defects, is crucial when developing web or mobile applications. Testing with automation is used in each of the JavaScript and Django coding processes to assist find and fix mistakes.
- **Quality Control and Testing:** During development, comprehensive testing and quality control procedures are employed to identify and address any possible problems or defects. Usability, efficacy, and user acceptance were all tested. One goal of security testing was to find and address vulnerabilities. Administrators and

stakeholders were consulted before the necessary changes were made to ensure a dependable, high-quality system.

10.3 Prioritization while developing

Prioritization	Requirements and Explanation
Core Functionality	The main features of an AI-powered web application for Chicken Disease Detection include effective image upload as well as management, precise disease identification and categorization, a clear display of results, an intuitive interface, strong security and confidentiality of information, incorporation with other systems, continuous model maintenance, and extensive user support. Together, these features enable improved farm management and chicken health by offering a dependable and user-friendly tool for identifying and treating poultry diseases.
UX	An intuitive interface that is commonplace for choosing disease-related photos utilizing pure information for all file types in the exact location of the computer. It is necessary to assess usability and collect feedback in order to consistently provide high-quality usability.
Security and Data Management	Durable user access controls, secure authentication, and robust data encryption. Assess the web application with real users to obtain their feedback to continuously improve its usability. Protecting user data and preserving data integrity should receive serious attention. Adequate access restrictions, secure user authentication, and robust data encryption should all be carefully considered in order to maintain data security and protect admin information about users.
Optimization Performance	The web application's efficiency was increased through performance optimization, which included speeding up effective caching methods and increasing the network's coding efficiency.
Integration with DL models with accuracy	Appling 4 models of DL for get the best predict with accuracy which can help for generate a web application for classify 4 classes of data.
Quality Assurance and Testing	I'm developing an web application at the moment. I have a lot of code to develop for this project. There are a few mistakes in the code. I employed automated testing to find a solution to this issue. The quality of the project is confirmed by automation testing. Therefore, look for and identify errors prior to the planned release.

User Feedback and Continuous Improvement	Every program requires user input since it improves overall software functionality and aids in the development team's understanding of novel concepts. It's critical to fix errors and work to enhance the program's functioning when feedback is released.

CHAPTER 11

Testing

Project Name	Chicken disease detection web application using AI		
Name of product	Chicken disease detection web application using AI Web App		
Product description	Chicken disease detection web application using AI Web App		
Project description	JavaScript, Django, Python, MySQL.		
Project duration	Project Type	Testing/ Verification	
	Start date	End date	

11.1 Test Plan Acceptance

List the goals and limitations of the exam. Determine the important participants and get their consent before implementing the test plan. Indicate each testing step's acceptance criteria in clear terms.

11.2 Unit Testing

Unit tests should be performed by combining the structure as a whole and testing each module independently. As the smallest component of each module, the software's architecture serves as a focal point for verification efforts thanks to unit testing. An

alternative word for this is module testing. Every system module is looked at separately. Make sure this technique works with all browsers as well.

11.3 Validation Testing

Software testing employs processes for validation and verification to ensure that a system satisfies requirements and operates as intended. It might also be known as quality assurance for software.

11.4 Integration Testing

The problems around the two concerns related to inspection and program creation are addressed by integration testing. Following software integration, a series of high-order evaluations are run. This testing strategy's primary goal is to construct a program structure in accordance with design requirements by utilizing unit-tested components.

11.5 TEST CASES

Table 7: Test Case

Case Id	CASE NAME	Expected Result	Actual Result	Result (Pass/Fail)
1	Choose diseases images	Click button of choose images then give the access to select of computer images.	After clicking button of choose images then give the access to select of computer images.	Pass
2	Detection chicken diseases.	Detection 4 classes of disease using chicken coops images.	Detection 4 classes of disease using chicken coops images accurately.	Pass

CHAPTER 12

Implementation

12.1 Training

User	Training	Time	Comment
Users or Clients			

12.2 Big Bang Implementation

The Big Bang approach to execution deploys the full system at once and does not use a trial or parallel process of implementation beforehand. Entire instructions must be given to all web app users, especially the admin, for the purpose of planning the detection of chicken illnesses. I began experimenting with the coding before attempting out the concept. I assessed the precision of each of the four algorithms that were employed. After everything was finished, I evaluated the process' accuracy. I determined which would work best for my needs after weighing the accuracy. Statistics on the conditions of chicken coops have showed this to be quite precise in terms of detecting diseases in chickens. A comprehensive analysis of all relevant mathematical and philosophical ideas and techniques led to the development of a set of basic conditions that must be satisfied for each and every effort at picture categorization.

12.3 Scaling

In order to manage increasing load, guarantee speed, and preserve accuracy as an increasing amount of users and data develops, scaling an AI-powered web app for chicken illness diagnosis entails taking care of a variety of issues. Infrastructure optimization, AI model enhancement, effective data management, and reliable application performance are all necessary for scaling a web application for detecting chicken diseases. Optimizing machine learning algorithms for interpretation and training, maintaining scalable networks and storage, concentrating on security and performance,

and utilizing services in the cloud, containerization, which is and load balancing are important tactics. The system's ability to adapt to changing demands and maintain effectiveness is ensured by ongoing improvement methods such as model updating and user input.

12.3.1 Design of scaling

Multiple architectural layers and factors must be taken into account while creating a scalable AI online application for the diagnosis of chicken sickness. Using AI to scale a web application for detecting chicken disease requires building an efficient framework that can withstand loads, retain accuracy, and guarantee outstanding performance.

12.3.2 Testing Performance

A little program has to be created for each task the scanner does, such as disclosure, susceptibility detection, static and dynamic assessment, and analysis. Teach the architectural and development teams how to create systems that are both scalable and effective. This entails being aware of elements such as database efficiency, caching, horizontal scalability, and performance monitoring.

12.4 Experiment Result

Algorithms have predicted the discovery of many. sicknesses in chickens based on pictures of sick coop chickens. As a result, I employed many tactics. I considered and assessed a range of options before deciding on the best course of action for the trial. In an effort to improve the caliber of my work, I tried a number of different approaches. used Kaggle's public datasets for the four chicken illnesses that were being taught. I made advantage of the pre-existing Python tools, dictionaries, and content classification strategies. This study employs Python DL modules for identification to apply deep learning once more to the task of classifying images of chicken diseases into 4 categories. developing a web-based application to determine whether different chicken coop environments can be recognized from photos of chicken coops.

12.4.1 Descriptive Analysis

My methods of categorizing affected the results that I got. I have pinpointed the precise locations of chicken diseases in photographs using four distinct deep learning methods. I use deep learning methods (CNN, VGG16, MobileNetV3, & Resnet) that have demonstrated potential for precise illness diagnosis in chickens. All the models shared the same the data set, which comprised publicly available data and my own online dataset sourced from Kagle sources. After finishing the dataset method, I assessed the methods' soundness using the pre-made libraries in Mat-lab. I also used online prototypes to forecast chicken diseases.

Table 8: Accuracy Table

Models	**Accuracy Score (AUC)**
CNN	43.68%
VGG16	77.23%
MobileNetV3	70.50%
Resnet	81.70%

In this works, there have been 4 models have been used. The accuracy ratings (AUC) of different models show notable variations in their functionality. With an amazing score of 81.70%, ResNet, out of all the evaluated models, had the best accuracy, demonstrating its strong suit for the task at hand. up close pursuit, VGG16 turned up a strong effort, scoring 77.23%. With an overall rating of 70.50%, MobileNetV3 did rather well, demonstrating its efficiency and keeping respectable accuracy. Conversely, the CNN model performed poorly, scoring only 43.68%, indicating that it might not be as useful in this particular application. ResNet and VGG16 are clearly the best models overall, with the conventional CNN model doing poorly.

The next section displays the effectiveness for several models. Two open-source programs, PyCharm and CoLab, were used in the process. In all, four models were utilized: CNN, VGG16, MobileNetV3, and Resnet. With an accuracy of 81.70%, the Resnet model was the best-performing model.

Fig: Accuracy & validation curve

CHAPTER 13

Critical Appraisal and Evaluation

13.1 Objective that could be met

These goals can help in designing a web application for AI-based chicken illness diagnosis that meets high requirements for accuracy, achievement, and user happiness. A successful and significant application will include a mix of real-time processing, scalability, user-friendliness, thorough illness coverage, accurate disease identification, and strong security measures. The system's long-term efficacy and sustainability are guaranteed by regular assessment of performance, feedback integration, cost effectiveness, and regulatory compliance.

13.1.1 Success rate against each objective

By establishing precise, quantifiable goals and routinely evaluating performance in relation to these goals, rates of achievement for each aim are ascertained. Fulfilling these goals guarantees that the AI-powered online application for detecting chicken diseases is precise, scalable, safe, and dependable for its customers.

13.1.2 How much better could have been done

The model preciseness, immediate execution rapidity, adaptability, accessibility, data administration, model flexibility, cost effectiveness, and regulatory compliance are all important factors to take into account while making improvements to the AI-powered web app for detecting chicken diseases. The app will be more effective and have a greater influence on disease detection if these enhancements are put into practice. It will also function better, offer a better user experience, manage bigger loads, and maintain strong compliance and safety measures.

13.1.3 Why it could not be done

Even though AI has a lot of potential to improve the identification of poultry diseases, there are a number of obstacles that might prevent it from being used effectively. These include restrictions on the availability and quality of data, processing power, technological intricacy, compatibility with current platforms, adherence to regulations, and financial limits. A multidimensional strategy, involving enhanced data collecting, sophisticated model development, and successful stakeholder involvement, is needed to address these issues. To overcome these obstacles and fully utilize AI in this field, constant innovation and adaptability are essential.

13.1.4 Which objectives have been missed

A web app that uses AI to identify chicken sickness missed its goals for a number of reasons, such as problems with integration and user experience, data-related problems, technological limits, and resource shortages. To close these gaps, significant work must be put into ensuring scalability, optimizing user interface design, enhancing real-time processing, improving model accuracy, and upholding stringent requirements for data security, tracking of performance, and complying with regulations. The efficacy and effectiveness of the AI-based illness detection system may be greatly increased by recognizing and addressing these issues.

13.1.5 Why these objectives have missed

The reasons behind the unfulfilled goals of an AI-powered online application for detecting chicken sickness include issues with data availability, computational limits, technological difficulties, budgetary restrictions, and regulatory obligations. It is frequently necessary to combine enhanced data procedures, cutting-edge technological solutions, and efficient resource management to address these problems. The efficacy and effectiveness of the AI-based illness detection system may be greatly increased by recognizing and resolving these issues.

13.1.6 What could have been done to complete those objectives

The AI solution for the chicken disease identification web app's unmet goals includes enhancing model performance and data quality, streamlining immediate processing and scalability, creating an intuitive user interface, guaranteeing thorough disease coverage, and upholding continuous availability and security. The efficacy and consequences of the application may also be greatly increased by concentrating on cost-effectiveness, regulatory compliance, and efficient feedback integration. A mix of technological know-how, ongoing improvement initiatives, and strategic planning are needed to put these measures into action.

13.2 Objectives totally not met / touched

Many obstacles, such as resource shortages, regulatory hurdles, and technological limitations, could prevent immediate handling, thorough disease coverage, excellent availability and trustworthiness, cost effectiveness, complying with regulations, user feedback insertion, and operational monitoring from ever being achieved. Targeted approaches including extending disease coverage, modernizing infrastructure, enhancing compliance procedures, cutting expenses, and putting in place reliable feedback and tracking mechanisms are needed to close these gaps. The AI-based chicken illness detection system may be made much more successful and impactful by concentrating on these areas.

13.2.1 Including software and documentation

To ensure the efficacy and efficiency of the software, regular inspection, problem fixes, and user input would have been incorporated into the software development process. To ensure simple comprehension and future upgrades, the documentation—which comprised user manuals, technical directions, and service instructions—had to be extensive.

CHAPTER 14

Lessons Learned

14.1 Pre-project

Typically, I would describe the pre-project goals, limitations, and specifications for the online application that uses artificial intelligence (AI) to identify chicken diseases. This may entail determining the intended user population, carrying out market research, and deciding what features and functions the application needs. In addition, I might have to draft a project plan that includes objectives, deadlines, and a list of all the necessary technological and human resources. Developing a comprehensive project plan that includes personnel, software tools, and time limitations.

14.2 Review

To evaluate project progress, spot plan deviations, and make sure everything is moving forward on time, regular reviews are crucial. Using these assessments, I can assess the web application's success in accomplishing its objectives and decide whether any changes or enhancements are necessary. Regular input from readers and other participants can aid in pinpointing areas in need of improvement and facilitate well-informed decision-making.

14.3 Lessons Learned

I conducted a thorough investigation before creating this web application. I studied system design and analysis before I started working on this program. This is the most difficult phase of system development. It is impossible to generate precise programming without analysis. I then studied Python, Django, and JavaScript. Acquiring knowledge is challenging. Despite their widespread use, MySQL and JavaScript may be challenging to learn and utilize in the simplest versions. As soon as I begin learning, I become aware of my shortcomings so that I might be considered for next duties

14.4 Problem Faced

It takes skill to overcome complicated obstacles in the quality of data, accuracy of models, immediate processing, expansion, consumer experience, regulation compliance, and cost when creating an AI-powered web app for the identification of chicken illness. It will need a systematic strategy to overcome these obstacles, one that involves creating user-friendly interfaces, guaranteeing high availability, maximizing computational resources, and enhancing data quality. The efficiency and dependability of the program may be greatly increased by taking care of these issues.

14.5 Problems That are solutions

These software tasks are challenging. despite the fact that I create web apps. I encountered a few typical issues while working on the project, but these are not the really difficult ones. Coming up with ideas that would really work, however, could be challenging. By proactively addressing these concerns, websites for chicken diseases detection systems may improve client satisfaction, boost operational effectiveness, and position them as trustworthy sources in the competitive vetenary industry. Regular monitoring of user feedback and performance metrics will further inform ongoing improvements and optimizations.

CHAPTER 15

Conclusion

15.1 Summary of the project

The complete findings for my project, a "Chicken disease detection web app using AI," are available here. The steps used to turn the concept into a functional website are covered in full in this article. One component that stands out to system users is the user panel. Poultry illnesses are extremely detrimental to the operations of any poultry farm.Because there are little opportunities for agricultural support services available to chicken growers, many diseases do not manifest early indications in their flocks of birds. Deep learning techniques could be able to detect some poultry diseases at an early stage. To diagnose illnesses in chickens, this website will need user participation. It will thus consist of two fundamental parts. The first entails creating websites, and the second uses deep learning to detect illnesses in chickens.

15.2 Goal of the project

The main objective of this project is to create an easy-to-use web application that can accurately identify and classify hen diseases based on specific symptoms or images. By applying AI techniques, we want to provide farmers with a trustworthy and efficient tool that might aid in the early detection of illnesses, reduce the possibility of monetary losses, and improve animal welfare. The initiative seeks to improve the health of animals and efficiency of operations in poultry production by giving special attention to preciseness, user convenience, scalability, and compliance. It also hopes to offer farmers and vets a useful resource.

15.3 Success of the project

This project has achieved great success. The principal actions are:

- Detection chicken disease through coop of chicken images.
- Basic Info about company
- Contact Us page

15.4 Documentation

The following phases, tasks, and plans were probably included throughout the documentation:

- **Prior-Project Records**: Project concepts, viability analyses, and a preliminary needs gathering may fall under this category.
- **Project plan:** A written document that describes the goals, limitations, schedule, necessary materials, and risk-reduction tactics for a project.
- **Technical specifications:** a set of precise guidelines pertaining to the features, functionalities, and layout of an online application within a travel or tourist administration system.
- **User Documentation**: Provides advice and guidelines on how to use the program efficiently to all parties concerned, including that individual in charge who searches for holiday spots or packages.
- **Examination and Quality Assurance Instruction**: To make sure the software satisfies quality requirements, document test strategies, scenarios, and outcomes.
- **Deployment and Maintenance plans**: Plans are supplied for the deployment, regular consumption, and updating of applications, as well as documentation detailing the required actions.

15.5 Value of the project

Artificial intelligence (AI)-powered Chicken Disease Identification Web Apps have many uses, including cost savings, improved health, tactical benefits, and educational gains. The program promotes innovation in the chicken sector, increases public health, boosts production, and improves animal welfare by offering prompt and accurate illness diagnosis. All of these advantages help make chicken farming operations more successful and sustainable overall.

15.6 My Experience

I may contribute my software creation, design, or project management expertise to the tourist management system web application project. My experience has given me the ability to collaborate with stakeholders, effectively manage project schedules, overcome technological obstacles, and put into practice solutions created especially to predict chicken diseases. In order to advance my professional skills, I have also strengthened my problem-solving, teamwork, and documentation skills.

References

[1] D. Machuve, E. Nwankwo, N. Mduma and J. Mbelwa, "Poultry diseases diagnostics models using deep learning" *Frontiers in Artificial Intelligence*, *5*, p.733345, 2022.Duckett, J., 2001. *Web Design with HTML, CSS, JavaScript and jQuery*. 1st ed.

[2] W3schools.com. 2021. *JavaScript Tutorial*. [Online] Available at: <https://www.w3schools.com/js/default.asp> [Accessed 17 December 2021].

[4] W3schools.com. 2021. *What is Bootstrap*. [Online] Available at: <https://www.w3schools.com/whatis/whatis_bootstrap.asp> [Accessed 17 December 2021].

[5] Learn about React Js, available at <<https://youtu.be/4UZrsTqkcW4>>, Last accessed on 20-01-2023 9:48 PM.

[6] Learn about Node Js, available at <<https://nodejs.org/en/>>, Last accessed on 15-01-2023 5:00 AM.

[7] Learn about HTML and CSS, available at <<https://youtu.be/-8ORfgUa8ow>>, Last accessed on 04-012023 4:23 AM.

[8] Learn about JavaScript, available at <<https://youtu.be/2Ji-clqUYnA>>, Last accessed on 21-01-2023 5:08 AM.

[9] G. Franzo, M. Legnardi, G. Faustini, C. M. Tucciarone, and M. Cecchinato, "When everything becomes bigger: big data for big poultry production," Animals, vol. 13, no. 11, p. 1804, May 2023, doi: 10.3390/ani13111804.

[10] X. Li et al., "The genetic architecture of early body temperature and its correlation with salmonella pullorum resistance in three chicken breeds," Frontiers in Genetics, vol. 10, Jan. 2020, doi: 10.3389/fgene.2019.01287.

Hasib

ORIGINALITY REPORT

6%	4%	0%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	Submitted to University of Wales, Lampeter Student Paper	<1%
3	Submitted to University of Greenwich Student Paper	<1%
4	Submitted to University of Southern Mississippi Student Paper	<1%
5	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
6	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
7	Submitted to University of Wolverhampton Student Paper	<1%
8	digital.library.unt.edu Internet Source	<1%
9	ijeecs.iaescore.com Internet Source	<1%