

DeshiTalent – Bangladeshi Freelancers Hub (DTBFH)

BY

Md. Ruhul Kuddus
ID:211-16-555

This Report was Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Computing and Information System (Major AI in IoT)

Supervised By

Md. Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing and Information System
Daffodil International University



**DEPARTMENT OF COMPUTING AND INFORMATION SYSTEM
FACULTY OF SCIENCE AND TECHNOLOGY
DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH**

12 JANUARY 2025

APPROVAL

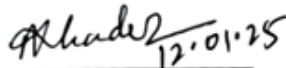
This Project titled “**DeshiTalent - Bangladeshi Freelancers Hub (DTBFH)**”, Submitted by. **Md. Ruhul Kuddus**, Student ID No: **211-16-555** to the Department of Computing and Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of **B.Sc. in Computing & Information Systems (Major AI in IoT)** and approved as to its style and contents. The presentation has been held on 12-01-2025.

BOARD OF EXAMINERS


12-01-25

Md Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Chairman


12.01.25

Md. Nasimul Kader
Assistant Professor
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner


12.01.25

Md. Mehedi Hassan
Lecturer (Senior Scale)
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Anowar Hossain
Founder & CEO at DocTime

External Examiner

Declaration

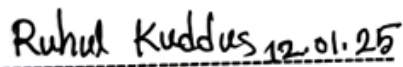
I hereby declare that this project has been done by me under the supervision of **Md. Sarwar Hossain Mollah, Associate Professor and Head** of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By

 12.01.25

Md. Sarwar Hossain Mollah,
Associate Professor and Head
Department of Computing and Information System (CIS)
Daffodil International University

Submitted By

 12.01.25

Md. Ruhul Kuddus
ID: 211-16-555
Department of Computing and Information System (CIS)
Daffodil International University

Acknowledgement

I would like to express my deepest gratitude to Allah, the Almighty, for granting me the strength and ability to successfully complete this project and its accompanying documentation. I am sincerely thankful to my parents and family for their unwavering support, encouragement, and understanding throughout the entire course. Their constant love and motivation have been pivotal in enabling me to reach this achievement. I extend my heartfelt thanks to my project supervisor, Md. Sarwar Hossain Mollah, for her invaluable guidance, assistance, and consistent support during not only this course but also my whole undergraduate life. Her expertise and advice have greatly contributed to the successful completion of this work. On the other hand, Md. Nasimul Kader and Md. Mehedi Hassan also gratefully for their adviser and support Finally, I wish to convey my appreciation to all my professors at Daffodil International University, my friends, and well-wishers who have supported and encouraged me from the beginning of this project. I am truly grateful to each one of them for their contribution to my academic journey.

Dedication

I like to dedicate this project to my parents (Md. Abdur Rahim and Mis. Rona Begum). Without my parents' courage, support and inspiration I am nothing but incomplete. Their encouragement, wisdom, and belief in me have been instrumental in shaping my journey. This achievement reflects their continuous guidance and the values they have instilled in me. I am forever grateful for their presence in my life, and this accomplishment is dedicated to them with all my love and respect.

Executive Summary

DeshiTalent – Bangladeshi Freelancer Hub (DTBFH) is a web-based platform designed to connect Bangladeshi freelancers with clients, addressing challenges such as payment barriers, trust issues, and the lack of cultural and linguistic relevance in existing global platforms. The platform provides a secure and localized freelancing environment, empowering skilled professionals to showcase their expertise and access diverse opportunities. Built using Python, Django, SQLite3, TailwindCSS, and REST APIs, DeshiTalent ensures a scalable, user-friendly experience. It features SSLCommerz integration for seamless payment processing, catering to local needs. Bilingual support (Bangla and English) enhances accessibility for a diverse audience. The project fosters trust through transparent profiles, ratings, and reviews while creating economic opportunities and supporting skill development. By addressing critical gaps in the freelancing ecosystem, DeshiTalent aims to strengthen Bangladesh's position in the global freelancing market and contribute to local economic growth.

Table of Contents

APPROVAL.....	ii
Declaration.....	iii
Acknowledgement	iv
Dedication.....	v
Executive Summary	vi
List of Table	x
List of Figures.....	xi
Chapter 1 - Introduction.....	1
1.1 Introduction	1
1.2 Content of the Document.....	1
Chapter 2 - Preliminary Stages	4
2.1 Proposal Project Overview	4
2.3 Background of the project	4
2.3 Feasibility of Project.....	5
Chapter 3 - Literature Review.....	10
3.1 Introduction to Freelancing Platforms	10
3.2 Problem Domain of Background Study.....	10
3.2.1 Local Freelancing Platforms in Bangladesh.....	11
3.2.2 Overview and Local Marketplaces: Need and Scope.....	11
3.2.3 Scope of Local Marketplaces	11
3.3 Problems with the Existing Systems.....	12
3.4 Solution to the Problem	12
Chapter 4 - Methodology	14
4.1 Methodology.....	14
4.2 Suitable method	14
4.2.1 Agile Software.....	14
4.2.2 Waterfall Model:.....	16
4.2.3 DevOps.....	17
4.3 Choosing Methodology	18
4.3.1 Suitable my project.....	18
4.3.2 Sections of the Methodology.....	19
Chapter 5 - Planning	20

5.1 Project Plan.....	20
5.1.1 Working Structure Breakdown.....	20
5.1.2 Resource Allocation.....	21
5.1.3 Time Boxing.....	22
5.1.4 Gantt Chart.....	23
5.2 Test Plan.....	23
5.2.1 Testing Against the Time Boxes.....	24
5.2.2 Required Test.....	25
5.3 Risk Management.....	26
5.4 Change Management Approach.....	28
5.5 Standard Quality Measures.....	29
Chapter 6 - Feasibility Analysis.....	30
6.1 Possible feasibility analysis.....	30
Chapter 7 - Foundation.....	32
7.1 Problem Area Identification.....	32
7.1.1 Interview.....	32
7.1.2 Observation.....	33
7.2 Implemented technology.....	33
7.8 Recommendation and Justification.....	35
Chapter 8 - Exploration.....	36
8.1 Full Use of System.....	36
8.2 Activity Diagram of System.....	37
8.3 Requirements Catalogue.....	38
8.4 Prioritized Requirements List (PRL).....	39
8.4 Prototype of the new system.....	41
Chapter 9 - Engineering.....	47
9.1 Use Case Diagram.....	47
9.2 ER diagram of DTBFH.....	48
9.2 Sequence Diagram of DTBFH.....	49
9.3 Full System Activity Diagram.....	50
9.4 Logical Diagram of MVT Architecture.....	51
Chapter 10 - Testing.....	52
10.1 Functional Testing.....	52

10.1.1 Unit Testing	52
10.1.2 Module Testing.....	53
10.1.3 Integration Testing.....	53
10.2 Non-Functional Testing	54
10.3 Unit Testing	54
10.4 Module Testing	55
10.4 Acceptance Testing	55
10.5 Integration Testing	61
Chapter 11 - Deployment	62
10.1 Core module code snippet	62
Chapter 12 - Implementation	70
12.1 Process of Implementation	70
Chapter 13 - Critical Appraisal and Evaluation	71
13.1 Objective.....	71
13.2 Objective Not Fully Achieved.....	72
Chapter 14 - Lessons Learned.....	73
14.1 Pre-project Review Closing.....	73
14.2 What I have Learnt	73
14.3 The problems I have faced.....	74
14.4 What Solution Occurred	74
Chapter 15 - Conclusion	75
15.1 Summary of the Project	75
15.2 Project Goal	75
15.3 Success of the Project.....	75
15.4 Documentation Contribution	76
15.5 Value of the Project.....	76
15.6 Personal Experience	76
References.....	77
Plagiarism Report.....	78

List of Table

TABLE 2.1: TECHNOLOGY TO USED WEB-BASED APPLICATIONS OF DESHITALENT	6
TABLE 2.2: SYSTEM CONFIGURATION.....	6
TABLE 2.3: COST BENEFIT ANALYSIS OF FIVE YEARS.....	7
TABLE 2.4: KEY ACTIVITIES OF PROPOSED DESHITALENT.....	8
TABLE 2.5: MOSCOW PRIORITIZATION OF DESHITALENT PROJECT	9
TABLE 5.1: TESTING AGAINST THE TIME BOXES.....	24
TABLE 5.2: TESTING AND TEST PLAN ACTIVITY.....	24
TABLE 5.3: REQUIRED TEST OF PROPOSED PROJECT.....	25
TABLE 5.4: TEST CASES FORM	26
TABLE 5.5: USER ACCEPTANCE TEST PLAN FORM	26
TABLE 5.6: CONSIDERING BUSINESS PRIORITIES.....	29
TABLE 6.1: TECHNOLOGY TO USED OF PROPOSED DESHITALENT SYSTEM	31
TABLE 6.2: SYSTEM CONFIGURATION.....	31
TABLE 8.1: REQUIREMENTS CATALOGUE OF DESHITALENT PROJECT.....	39
TABLE 8.2: PRIORITIZED REQUIREMENTS LIST OF DESHITALENT.....	40
TABLE 10.1: UNIT TEST DETAILS.....	54
TABLE 10.2: MODULE TEST DETAILS.....	55
TABLE 10.3: ACCEPTANCE TEST DETAILS	55
TABLE 10.4: FREELANCER REGISTRATION TEST CASES PLANNING	56
TABLE 10.5: CLIENT JOB POSTING TEST CASES PLANNING.....	56
TABLE 10.6: PAYMENT PROCESSING TEST CASES PLANNING	57
TABLE 10.7: PROPOSAL SUBMISSION OF TEST CASES	58
TABLE 10.8: USER REGISTRATION AND LOGIN OF TEST CASES.....	58
TABLE 10.9: MODULE OF PROPOSAL SUBMISSION TEST CASES	59
TABLE 10.10: MODULE OF PAYMENT PROCESSING TEST CASES	60
TABLE 10.11: MODULE OF FREELANCER AND JOB SEARCH TEST CASES	60
TABLE 10.12: INTEGRATION TESTING OF USER REGISTRATION AND LOGIN INTEGRATION TEST STEP.....	61
TABLE 10.13: INTEGRATION TESTING OF JOB POSTING AND PROPOSAL SUBMISSION INTEGRATION.....	61
TABLE 12.1: PROCESS OF IMPLEMENTATION.....	70

List of Figures

FIG 4.1: AGILE METHODOLOGY	15
FIG 4.2: WATERFALL MODEL	16
FIG 4.3: DEVOPS MODEL	17
FIG 5.1: WORK BREAKDOWN STRUCTURE	20
FIG 5.2: RESOURCE ALLOCATION OF PROPOSED DESHITALENT PROJECT	21
FIG 5.3: TIME BOXING CHART	22
FIG 5.4: GANTT CHART	23
FIG 6.1: COST BENEFIT ANALYSIS OF PROPOSED SYSTEM.....	31
FIG 7.1: SERVER – CLIENT-SIDE TECHNOLOGY	34
FIG 8.1: USE CASE DIAGRAM OF DESHITALENT	36
FIG 8.2: ACTIVITY DIAGRAM OF DESHITALENT	37
FIG 8.3: LOGIN PAGES.....	41
FIG 8.4: SIGNUP PAGES PART-1	41
FIG 8.5: SIGNUP PAGES PART-II	42
FIG 8.6: SIGNUP PAGES PART-III.....	42
FIG 8.7: AUTHENTICATION PAGE.....	43
FIG 8.8: DASHBOARD OF FREELANCER	43
FIG 8.9: SHOW PROJECT LIST	44
FIG 8.10: BOOKMARK PAGES.....	44
FIG 8.11: SHOW THE REGISTRATION PROCESS	45
FIG 8.12: CLIENT POST THE PROJECT	45
FIG 8.13: PROJECT DETAILS SHOW	46
FIG 9.1: USE CASE DIAGRAM OF DESHITALENT	47
FIG 9.2: ER DIAGRAM OF DESHITALENT	48
FIG 9.3: ACTIVITY DIAGRAM OF DESHITALENT	49
FIG 9.4: FULL SYSTEM ACTIVITY OF DESHITALENT.....	50
FIG 9.5: LOGICAL DIAGRAM OF DESHITALENT	51
FIG 10.1: AFTER JOB POST TEST	57
FIG 10.2: PAYMENT PROCESSING TESTING.....	57

FIG 11.1: LOGIN CODE OF DJANGO (PYTHON-VIEW.PY).....	62
FIG 11.2: CLIENT PROFILE CODE OF DJANGO (PYTHON-VIEW.PY)	63
FIG 11.3: FREELANCER OF DJANGO (PYTHON-VIEW.PY)	64
FIG 11.4: DASHBOARD CODE OF DJANGO (PYTHON-VIEW.PY).....	65
FIG 11.5: HOMEPAGE CODE OF DJANGO (PYTHON-VIEW.PY)	66
FIG 11.6: AUTHENTICATION CODE OF DJANGO (PYTHON-VIEW.PY).....	67
FIG 11.7: POSTING PROJECT CODE OF DJANGO (PYTHON-VIEW.PY).....	67
FIG 11.8: CLIENT MANAGEMENT PROJECT CODE OF DJANGO (PYTHON-VIEW.PY)	69

Chapter 1 - Introduction

1.1 Introduction

DeshiTalent - Bangladeshi Freelancer Hub is a robust platform designed to connect freelancers and clients, addressing the challenges of talent sourcing and project management in Bangladesh. This initiative aims to establish an efficient marketplace that addresses these critical needs. In an era where digital transformation is reshaping work dynamics. The **DeshiTalent** project focuses on creating an efficient marketplace for freelancers and clients to collaborate seamlessly.

Key features include a robust user management system, streamlined project posting and bidding processes, secure communication channels, and secure payment solutions integrated with SSLCommerz and streamlined project management. Leveraging cutting-edge technologies such as Python, Django, SQLite3, and Tailwind CSS. DeshiTalent offers a scalable and user-friendly interface. With freelancers often struggling to find reliable opportunities and clients facing difficulty in sourcing competent talent, DeshiTalent bridges this gap effectively.

DeshiTalent aims to empower Bangladeshi freelancers and client, fostering a digital ecosystem where local talent thrives, and project execution becomes hassle-free, efficient, and secure.

1.2 Content of the Document

The project work in the upcoming chapters is described in depth in this document.

Chapter 1: Introduction

This section gives a brief overview of the suggested system and the overall project structure.

Chapter 2: Preliminary Stage

This chapter presents an initial investigation into the proposed system. It encompasses a delineation of its primary goals and objectives, a thorough examination of the identified problem area, an exploration of potential alternative solutions, and a relevant overview of the project's background

Chapter 3: Review of Literature

The chapter, the problem area is thoroughly discussed, possible solutions are analyzed, current solutions are assessed, and recommendations are made.

Chapter 4: Methodology

This chapter encompasses the rationale for the chosen approach, a critical evaluation of potential methodologies, and a detailed exposition of the selected methodology and its implementation.

Chapter 5: Strategic Planning

This chapter provides a thorough analysis of the many projects planning documents, including the test plan, the project plan, and the strategies for risk and change management.

Chapter 6: Feasibility

This section presents a comprehensive cost-benefit analysis and a thorough feasibility study report.

Chapter 7 - Foundation

This chapter presents an identification of the problem domain, outlines general requirements, explores suggested technologies, and provides detailed explanations of each.

Chapter 8 – Exploration

UML diagrams, class diagrams, a requirements catalog, prototypes detail in this chapter's thorough analysis of the current and suggested systems.

Chapter 9 –Engineering

In this chapter to discussion logical and behavioral models, and system architecture are all covered analysis of the current and suggested systems.

Chapter 10: Deployment

This section discusses the coding samples and the separation of development problems by development priority.

Chapter 11: Testing

The results of several test designs are documented in this chapter.

Chapter 12: Implementation

This chapter provides a study of the implementation strategy, the suggested implementation system, and associated factors.

Chapter 12: Critical Assessment and Evaluation

This chapter assesses the initial aims, distinguishing between accomplished and unaccomplished targets.

Chapter 13: Lessons Learned

This chapter discusses a key part of the pre-project ending review: looking at both the strategies that worked and the ones that didn't while the project was being carried out.

Chapter 14: Conclusion

This chapter provides a summary of the project, encompassing its objectives, accomplishments, and significant experiences.

Chapter 2 - Preliminary Stages

2.1 Proposal Project Overview

DeshiTalent is a web-based platform designed to connect Bangladeshi freelancers with clients, providing a centralized hub for managing freelance projects, bids, profiles, and payments. The platform empowers freelancers by offering a local marketplace to showcase their skills while allowing clients to find and hire top-tier talent for their projects. The purpose of DeshiTalent is following:

- A professional ecosystem for Bangladeshi freelancers.
- Create opportunities for clients to hire local talent conveniently.
- Provide a transparent, secure, and user-friendly platform for freelancing.

2.3 Background of the project

In a competitive job market, people pursue steady employment, whereas firms seek substantial support. Freelancer.com provides a worldwide employment network that links businesses and freelancers for mutual benefit (Freelancer.com, 2021).

In 2017, a survey by the Oxford Internet Institute (OII) indicated that Bangladesh was the second most prominent country for providing online labor. Forbes ranked Bangladesh among the top 10 countries globally for freelancing income in 2019. This growth in freelancing not only highlights the adaptability of the workforce but also underscores the importance of digital skills in today's economy. As more individuals turn to online platforms for employment, the demand for diverse skill sets continues to rise, creating opportunities for both seasoned professionals and newcomers alike.

The ICT Division estimates the local freelancing sector's value to be approximately \$1 billion. The "Digital Economy Report-2019" by UNCTAD indicates that Bangladesh is becoming a notable participant in the global freelancing industry, with over 650,000 freelancers generating over \$100 million in yearly international remittances. Last year, Payoneer positioned Bangladesh as the eighth among the top ten freelancing nations globally.

The freelance sector in Bangladesh has expanded significantly over the last decade, with an increasing number of individuals pursuing freelancing as a legitimate career choice. She ascribed this development to other factors, such as the accessibility of high-speed internet, the surplus of skilled professionals, and the rising need for remote labor resulting from the COVID-19 epidemic.

The current landscape of freelancing in Bangladesh reveals several challenges faced by both freelancers and clients. International freelancing platforms like Upwork and Fiverr charge significant service fees (often 10-20% per transaction), reducing freelancers' income. Difficulty in withdrawing earnings due to limited integration of local banks with global payment systems. Many Bangladeshi freelancers face challenges when working with international clients due to language proficiency, cultural misunderstandings and secure easy to accessible payment system. To my proposed system to overcome the problem.

2.3 Feasibility of Project

Operational feasibility

The operational feasibility pertains to the usability, ease of use, and necessary functionalities of the proposed system. DeshiTalent incorporates adequate validation and verification for user inputs, facilitating seamless communication between clients and freelancers, ensuring that software developers (frontend and backend), UI/UX designers, and QA engineers are readily accessible in Bangladesh. Ultimately, securing payment transactions with local payment gateways will streamline transaction management, hence minimizing delays or obstacles in fund transfers.

Technical Feasibility

A technical feasibility assessment of "DeshiTalent – Bangladeshi Freelancer Hub" indicates that users can efficiently manage their tasks through the system, representing a significant improvement over previous manual method. Development to using current and widely adopted web and mobile technologies, the system is designed for broad compatibility, functioning effectively across various web browsers even with limited internet connectivity. Secure data access is ensured through a hierarchical access control system. Furthermore, the web-based application's platform independence, coupled with the affordability and cost-effectiveness of the chosen development platform and resources, confirms its technical feasibility.

Category	Component
Hardware	HP laptop (Configuration) Wi-fi Router
Software	Microsoft office Google Chrome Browser Windows 11 (Operating system) Visual Studio
Database	SQLite3
Technology	<u>Clint Side:</u> I. Html II. CSS III. JavaScript IV. Tailwind <u>Server Side:</u> I. Python (Version - 3.10) II. Django (Version - 5.1)

Table 2.1: Technology to used web-based applications of DeshiTalent

There are several options for developing the suggested system, including web-based applications, mobile applications, and desktop applications.

Economic Feasibility

Cost of Web based Application:

The complete application and data are hosted on a remote server, accessible via the Internet at any time from any location, with no installation required.

Equipment	Cost per Unit	Total Cost
Extranet network with VPN	₹ 3000 per month	₹ 3,000
Desktop PC (Core i9-12 ^h Gen, 3.6 GHz processor, 32 GB DDR4 RAM, 2 TB HDD)	₹ 1,50,000	₹ 1,50,000
Web, File, and Email servers	₹ 15,000 per month	₹ 15,000
Total		₹ 1,68,000

Table 2.2: System Configuration

Cost Benefit Analysis:

SL No.	Equipment	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Web based application cost	1,68,000	-	-	-	-	1,12,000
2	Domain and hosting Cost	25,000	25,000	25,000	25,000	25,000	1,25,000
3	Employee Expenses	40,000	40,000	40,000	40,000	40,000	2,00,000
4	Other Cost	25,000	25,000	25,000	25,000	25,000	1,25,000
	Total Cost	2,58,000	90,000	90,000	90,000	90,000	6,18,000

Table 2.3: Cost Benefit Analysis of five years

The **key activities** required to achieve the goals of my proposed project, **DeshiTalent - Bangladeshi Freelancer Hub**:

SL	Activity	Objective	Expected Outcome	Stakeholders Involved
1	Requirements Gathering	Identify functional and non-functional requirements from stakeholders.	Clear project scope and detailed feature specifications.	Project Manager, Clients, Freelancers
2	System Design	Develop architecture, database schema, and user interface designs.	Scalable and efficient system structure ready for development.	System Architects, UI/UX Designers
3	Backend Development	Implement core features using Python, Django, and SQLite3.	Functional backend with user management, project posting, bidding, and payment handling.	Backend Developers
4	Frontend Development	Build responsive user interfaces using TailwindCSS, HTML, and CSS.	Intuitive and user-friendly interfaces for freelancers, clients, and admins.	Frontend Developers
5	API Development	Develop REST APIs for seamless communication between the frontend and backend.	Smooth data flow and integration between platform components.	API Developers
6	Payment Gateway Integration	Implement SSLCommerz for secure escrow-based payment transactions.	Safe and reliable payment system for freelancers and clients.	Backend Developers
7	Testing and Quality Assurance (TQA)	Perform unit, integration, and system testing to detect and rectify errors.	Bug-free and secure platform ready for deployment.	QA Team

8	Beta Testing	Involve a select group of users to test the platform's usability and functionality.	Real-world feedback to refine the system.	Freelancers, Clients
9	Deployment	Deploy the platform on a live hosting environment with a CI/CD pipeline.	Fully functional platform accessible to users.	DevOps Engineers
10	Training and Documentation	Provide training materials and user guides for all stakeholders.	Enhanced user adoption and reduced learning curve.	Trainers, Technical Writers
11	Post-Deployment Support and Maintenance	Monitor performance, resolve issues, and release updates based on user feedback.	Continuous improvement and user satisfaction.	Support Team, Developers

Table 2.4: key activities of Proposed DeshiTalent

The **MoSCoW Prioritization** in a tabular format for **DeshiTalent - Bangladeshi Freelancer Hub**:

Priority	Feature	Description
Must-Have	User Authentication and Role Management	Secure login, registration, and role-based access for freelancers, clients, and admins.
	Project Posting and Bidding System	Allow clients to post projects and freelancers to submit bids.
	Secure Payment Gateway (SSLCommerz Integration)	Enable escrow-based payments for safe transactions between clients and freelancers.
	Messaging and File Sharing	Facilitate real-time communication and exchange of files between clients and freelancers.
	Dashboard for Freelancers and Clients	Provide personalized dashboards to track projects, payments, and progress.
	Admin Panel	Offer tools for platform management, dispute resolution, and analytics.
	Rating and Review System	Allow users to rate and review each other post-project completion to build trust and transparency.

Should-Have	Advanced Search and Filtering	Enable users to search and filter projects or freelancers based on specific criteria (e.g., skills, budget).
	Notifications System	Notify users about project updates, messages, payments, and other important activities.
	Dispute Resolution Module	Provide a system for resolving conflicts between freelancers and clients.
	Reporting and Analytics	Offer insights and analytics to admins, clients, and freelancers (e.g., earnings, project trends).
Could-Have	Skill Verification and Certification	Provide options for freelancers to verify skills or earn certifications.
	Freelancers' Portfolio Showcase	Allow freelancers to display a portfolio of past work on their profiles.
	Project Milestone Tracking	Allow clients and freelancers to set and track project milestones.
	Localization Features	Offer the platform in multiple languages (e.g., Bangla and English).
Won't-Have	Mobile Application for Location Tracking	Feature deferred due to time constraints and the need for live hosting servers.
	AI-Powered Talent Recommendation	Use AI to suggest freelancers for projects and vice versa (potential future enhancement).

Table 2.5: MoSCoW Prioritization of DeshiTalent Project

Chapter 3 - Literature Review

3.1 Introduction to Freelancing Platforms

Freelancing platforms have revolutionized the global job market by connecting clients and freelancers through online marketplaces. Platforms like Upwork, Fiverr, and Freelancer.com have enabled skilled professionals to offer their services on a global scale, contributing to the rise of the gig economy. According to the International Labour Organization (ILO), the gig economy has been growing exponentially, especially in developing countries like Bangladesh, where the freelancing workforce has become a significant contributor to the economy.

However, while global platforms provide opportunities, they often fail to address local market-specific needs, such as regional payment methods, language preferences, and market dynamics. This gap highlights the need for a localized freelancing platform that caters specifically to Bangladeshi freelancers and clients.

3.2 Problem Domain of Background Study

Global Freelancing Platforms

Global freelancing platforms such as Upwork, Fiverr, and Freelancer.com are well-established and provide a wide range of services. These platforms focus on:

- ✓ Connecting global clients with freelancers.
- ✓ Secure payment systems.
- ✓ Comprehensive project management tools.

However, these platforms have limitations for Bangladeshi freelancers:

- ✓ **Payment Barriers:** Many global platforms rely on international payment gateways, making it difficult for Bangladeshi users to withdraw earnings easily.
- ✓ **High Competition:** Freelancers from Bangladesh face stiff competition from professionals in other countries, often leading to underpricing.

- ✓ **Lack of Local Support:** These platforms do not provide support tailored to the local context, such as language or cultural nuances.

3.2.1 Local Freelancing Platforms in Bangladesh

Local platforms such as Dealancer, Freelancers.gov.bd, and Sheba.xyz have emerged to address the challenges faced by Bangladeshi freelancers on global platforms. These platforms provide:

- Local payment options (bKash, Rocket, and bank transfers).
- Services in Bangla for ease of use.
- Support for small and medium-sized enterprises (SMEs) seeking affordable freelancing services.

While these platforms address key pain points, they are still evolving and often lack the robust technological infrastructure and user engagement strategies seen in global platforms.

3.2.2 Overview and Local Marketplaces: Need and Scope

Freelancing is a vital part of the global gig economy, with Bangladesh emerging as one of the largest contributors. According to the Oxford Internet Institute, Bangladesh ranks among the top countries for online labor supply. Despite this achievement, the freelancing landscape in the country remains underdeveloped in terms of localized infrastructure.

Need for Local Marketplaces

Payment Accessibility:

Global freelancing platforms like Upwork, Fiverr, and Freelancer.com often rely on international payment gateways such as PayPal, which are not accessible in Bangladesh. Local platforms can bridge this gap by integrating popular payment options like **bKash**, **Rocket**, and local banks.

- **Cultural Relevance:**
Many Bangladeshi freelancers and clients prefer communication in Bangla and are more comfortable dealing with local platforms that understand their cultural context.
- **Expanding Local Opportunities:**
While global platforms cater to international clients, there is a growing demand for freelancing services within Bangladesh. Small businesses and startups seek affordable local talent, creating significant market opportunity.

3.2.3 Scope of Local Marketplaces

- **Economic Impact:**
The freelancing industry in Bangladesh contributes over **\$100 million annually**, with the potential for exponential growth if barriers are addressed.

- **Skill Development:**
Local platforms can encourage freelancers to develop their skills through mentorship programs and localized training resources.
- **Community Building:**
By creating a platform tailored to local needs, a freelancer hub can foster a sense of community and collaboration, boosting confidence and productivity.

3.3 Problems with the Existing Systems

Existing global and local freelancing platforms face numerous challenges that limit their effectiveness for Bangladeshi users:

Global Platforms

- ✓ **Payment Barriers:**
- ✓ **High Competition:**
- ✓ **Limited Local Relevance:**

Local Platforms

- ✓ **Technological Limitations:**
- ✓ **Trust Issues**
- ✓ **Fragmented Market**
- ✓ **Skill Development Gaps**

3.4 Solution to the Problem

DeshiTalent - Bangladeshi Freelancer Hub aims to address the challenges faced by freelancers and clients by providing a localized, technologically advanced, and user-friendly platform. The proposed solutions include:

Localized Payment Systems

- ✓ **Integration with Local Gateways:**
The platform will support local payment methods like **bKash**, **Rocket**, and bank transfers, making transactions seamless and cost-effective.
- ✓ **Transparent Fee Structure:**
A competitive fee model will ensure freelancers retain a higher percentage of their earnings.

Advanced Matching and Job Management

- **AI-Powered Recommendations:**
Using machine learning algorithms, the platform will match freelancers with suitable projects based on skills, experience, and client requirements.
- **Intuitive Job Management Tools:**
Freelancers and clients will benefit from integrated project tracking, milestone setting, and automated reminders.

Community and Skill Development

- **Mentorship Programs:**
Experienced freelancers can mentor newcomers, fostering a collaborative environment and improving overall skill levels.
- **Localized Training Resources:**
The platform will offer tutorials, webinars, and certifications in Bangla and English to help freelancers stay competitive.

Trust and Accountability

- **Secure Payment Escrow:**
An escrow system will ensure that payments are securely held until project milestones are met, protecting both freelancers and clients.
- **Dispute Resolution:**
A dedicated support team and structured dispute resolution mechanisms will build trust and encourage repeat usage.

Language and Cultural Relevance

- **Bangla Support:**
The platform will provide an interface in Bangla and English, making it accessible to a broader audience.
- **Cultural Customization:**
Features like tailored pricing models and project descriptions will reflect local business norms.

The review highlights the immense potential for a localized freelancing platform like **DeshiTalent**. While existing platforms provide essential services, they fail to address the unique challenges of the Bangladeshi market, including payment barriers, lack of cultural relevance, and limited technological capabilities. By addressing these issues, **DeshiTalent** can empower Bangladeshi freelancers, create new opportunities for local businesses, and contribute to the growth of the country's gig economy.

Chapter 4 - Methodology

4.1 Methodology

A project methodology is a structured framework for monitoring and carrying out projects. It establishes a framework for the definition of roles, responsibilities, and processes, thereby guaranteeing the efficient and effective completion of initiatives. Software development teams select methods carefully to improve the efficiency and success of their projects. Every technique has its own advantages and disadvantages, which makes it appropriate for a given project. A software development project's ability to succeed frequently depends on the choice of development approach.

4.2 Suitable method

Software development methodologies exert a significant influence on the effectiveness of the development lifecycle. Prominent examples of such methodologies include Waterfall, Agile, Prototype, Rapid Application Development (RAD), Dynamic Systems Development Method (DSDM), Spiral, and Joint Application Development (JAD). This section will delve into a detailed examination of three of these methodologies, with an emphasis on their respective strengths and weaknesses.

4.2.1 Agile Software

Agile software development prioritizes flexibility, collaboration, and client satisfaction. It is founded on the Agile Manifesto, which advocates for collective contribution, collaboration, the creation of functional software, and receptiveness to change. Agile is an iterative and incremental methodology, with the primary objective of delivering a functional product rapidly and frequently. This method emphasizes the necessity for the development team and the client to collaborate intimately to ensure the product fulfills all requirements and expectations.

Advantages of Agile Software Development

- ✓ **Flexibility and Adaptability:** Iterative methods in agile make it easy to respond quickly to changing market situations and customer needs.
- ✓ **Customer satisfaction:** Regular software delivery guarantees that client requirements are satisfied, and that input is considered at every stage of development.
- ✓ **Collaboration:** Working together closely with the customer and the development team leads to better communication and understanding.

- ✓ **Early Problem Finding:** Regular iterations and feedback loops help find and fix possible problems early on.
- ✓ **Higher Motivation:** The focus on providing value and the iterative approach can make the team more motivated and involved.

Disadvantages of Agile Software Development

- ✓ **Lack of Predictability:** Because iterative processes are so flexible, it can be hard to get exact estimates of project costs and timelines.
- ✓ **Chance of Scope Creep:** Because Agile is so flexible, changes in scope may happen without enough control.
- ✓ **Problems with Documentation:** Focusing on making software that works may result in not enough documentation, which can make upkeep and updates harder in the future.
- ✓ **Needs Highly Skilled Team Members:** For Agile to work, you need a team of highly skilled, driven people who can communicate well and solve problems.
- ✓ **Not Right for Every Project:** Agile might not be the best way to do some projects, like those with tight deadlines or detailed, complicated needs.

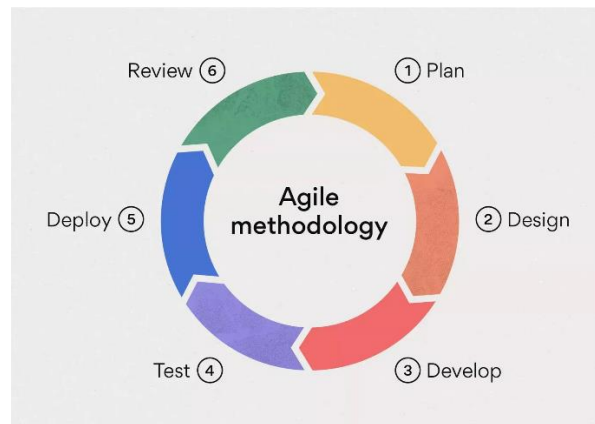


Fig 4.1: Agile methodology

Agile is an advantageous methodology that provides considerable benefits regarding adaptability, client pleasure, and cooperation. However, it is imperative to meticulously evaluate the potential disadvantages and guarantee that the team and project are well-suited for an Agile approach.

4.2.2 Waterfall Model:

- ✓ **Requirements gathering and analysis:** This stage entails recording the software requirements and comprehending the client's needs.
- ✓ **Design:** The requirements acquired are used to construct the software and system designs.
- ✓ **Implementation:** The program is conceived and coded.
- ✓ **Verification and Validation:** The program is examined to make sure it satisfies the requirements.
- ✓ **Deployment:** The software is implemented in the production environment.
- ✓ **Maintenance:** To address bugs and include new features, the program is updated and maintained.

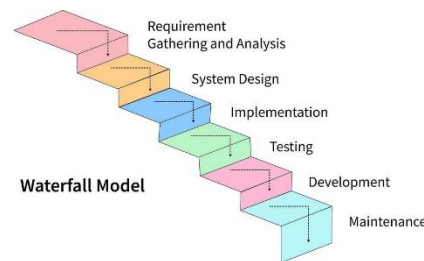


Fig 4.2: Waterfall Model

Advantages of the Waterfall Model:

- ✓ **Accessible and Comprehensible:** The model's linear structure facilitates understanding and management.
- ✓ **Clearly Defined Stages:** Each phase includes distinct deliverables and a review procedure, facilitating progress tracking.
- ✓ **Suitable for Minor tasks:** It is efficacious for minor tasks with well-defined parameters.
- ✓ **Beneficial for Risk Management:** Risks can be recognized and mitigated promptly during the development process.

Disadvantages of the Waterfall Model:

- ✓ **Inflexible:** Once a phase is finalized, it is challenging to revert and implement modifications, which might provide issues if requirements evolve.
- ✓ **Restricted Customer Engagement:** Insufficient customer feedback may result in a product that fails to satisfy their requirements.

- ✓ **Significant Risk of Failure:** Ambiguously articulated requirements at the outset may lead to project failure.
- ✓ **Inappropriate for Complex Projects:** It is unsuitable for intricate projects characterized by fluctuating requirements or extensive testing.

4.2.3 DevOps



Fig 4.3: DevOps Model

Key Principles of DevOps

- ✓ **Automation:** Automating repetitive tasks like building, testing, and deployment reduces errors and speeds up the process.
- ✓ **Collaboration:** Breaking down silos between development and operations teams promotes communication and shared responsibility.
- ✓ **Continuous Integration and Continuous Delivery (CI/CD):** Regularly integrating code changes into a shared repository and automating the build, test, and deployment process.
- ✓ **Monitoring and Feedback:** Continuously monitoring applications and infrastructure to identify and resolve issues quickly.
- ✓ **Infrastructure as Code:** Treating infrastructure as software, allowing it to be managed and provisioned using code.

Benefits of DevOps

- ✓ **Faster Delivery:** Reduced time to market for new features and updates.
- ✓ **Improved Quality:** Enhanced software quality through continuous testing and feedback.
- ✓ **Increased Reliability:** More stable and reliable applications with fewer outages.
- ✓ **Enhanced Collaboration:** Improved communication and teamwork between development and operations teams.
- ✓ **Greater Efficiency:** Automation and streamlined processes lead to increased efficiency.

4.3 Choosing Methodology

4.3.1 Suitable my project

A well-defined methodology is essential for ensuring that the project meets its requirements within the specified timeframe and budget. By adhering to a structured methodology, the development team can systematically address the project's objectives through proven, predefined steps. While selecting the ideal methodology for an academic project can be challenging, adopting a suitable framework significantly enhances the likelihood of successful completion and delivery.

4.3.2 Sections of the Methodology

The Agile methodology incorporates several key phases that the development team must follow to ensure effective project execution. These phases are outlined below:

1. **Pre-project Phase**

This phase involves the generation of the initial project idea. Key elements such as the project's time duration, budget, and preliminary requirements are identified and documented.

2. **Feasibility Study Phase**

The feasibility of the proposed project is evaluated across economic, technological, and operational dimensions to ensure its viability.

3. **Requirement Gathering Phase**

Functional and non-functional requirements are identified through various techniques during this phase.

4. **Requirement Analysis and Prioritization Phase**

Collected requirements are examined and prioritized using structured prioritization techniques, such as the MoSCoW method, to focus on critical features.

5. **Exploration and Engineering Phase**

This phase is the iterative core of the methodology. It involves exploring the requirements in detail and developing the solution iteratively to align with project goals.

6. **Review Phase**

Deliverables developed during the iterative process are tested with users for feedback. If changes are required, the project cycles back to the exploration and engineering phase for revisions.

7. **Implementation Plans**

The final stage of the methodology focuses on releasing the developed application for use. This phase includes identifying and resolving any issues, defining release criteria, and configuring the system for deployment. Once all criteria are satisfied, the new system is launched for operational use.

By following these structured phases, the Agile methodology ensures that project deliverables are systematically developed, reviewed, and implemented, thereby increasing the likelihood of achieving the desired outcomes.

Chapter 5 - Planning

5.1 Project Plan

This section demonstrates the planning process for project completion. In basically, the project is broken up into a few distinct components, and everything will be finished within the time frame that is determined upon during this phase if the planning is followed.

5.1.1 Working Structure Breakdown

This phase focuses on dividing the project into smaller, more manageable tasks. This methodical approach ensures efficient task allocation and facilitates timely completion within the established timeframe. By providing a clear framework, it enables accurate time estimation and effective resource distribution. Without such a structured approach, managing and completing the project could become considerably more difficult.

In the Work Breakdown Structure (WBS), the proposed system is organized into categories and subcategories, as illustrated in the chart below.

Task Name	Duration	Start Date	End Date
Introduction	7	10/15/2024	10/21/2024
Initial Study	7	10/22/2024	10/28/2024
Literature Review	5	10/29/2024	11/2/2024
Methodology	5	11/3/2024	11/7/2024
Planning	5	11/8/2024	11/12/2024
Feasibility Study	4	11/13/2024	11/16/2024
Foundation	3	11/17/2024	11/19/2024
System Analysis and Design Architecture	20	11/20/2024	12/9/2024
Deployment	15	12/10/2024	12/24/2024
Testing	3	12/25/2024	12/27/2024
Implementation	2	12/28/2024	12/29/2024
Critical Appraisal & Evaluation	1	12/30/2024	12/30/2024
Lessons Learned	1	12/31/2024	12/31/2024
Conclusion	1	1/1/2025	1/1/2025
Review	1	2/1/2025	2/1/2025
Total	80 Days		

Fig 5.1: Work Breakdown Structure

5.1.2 Resource Allocation

All assets and resources are allocated and managed systematically to ensure the successful completion of the proposed project. Resource allocation is a critical aspect of project planning, as it directly impacts the project's progress and efficiency. Being an academic project, I will assume multiple roles at different stages of development, as there is no dedicated team. To adhere to the pre-scheduled task delivery deadlines, the resource allocation plan for the *DeshiTalent – Bangladeshi Freelancers Hub (DTBFH)* project is outlined as follows:

SL	Task	Resource
1	Introduction	Analyst, User
2	Initial Study	Analyst
3	Literature Review	Analyst
4	Methodology	Analyst, User
5	Planning	Analyst, Team Leader
6	Feasibility	Analyst, User
7	Foundation	Analyst, Designer, Developer
8	System Analysis and Design Architecture	Analyst, Designer, Developer
9	Deployment	Designer, Developer
10	Testing	Developer, Tester, User
11	Implementation	Analyst, Developer, User
12	Critical Appraisal & Evaluation	Analyst
13	Lessons Learned	Analyst, Designer, Developer, Tester
14	Conclusion	Analyst

Fig 5.2: Resource Allocation of Proposed DeshiTalent Project

5.1.3 Time Boxing

The Agile Development Methodology segments tasks into specific timeboxes to ensure goals are achieved within the established deadlines. This approach is a crucial component of the project planning for *DeshiTalent – Bangladeshi Freelancer Hub (DTBFH)*. In this phase, all tasks are allocated to designated timeboxes with predefined durations. The iterative nature of the methodology requires that these tasks be completed within the assigned timeframes.

Timebox	Task	Resource
TB1	Introduction	Analyst, User
TB2	Initial Study	Analyst
TB3	Literature Review	Analyst
TB4	Methodology	Analyst, User
TB5	Planning	Analyst, Team Leader
TB6	Feasibility	Analyst, User
TB7	Foundation	Analyst, Designer, Developer
TB8	System Analysis and Design Architecture	Analyst, Designer, Developer
TB9	Deployment	Designer, Developer
TB10	Testing	Developer, Tester, User
TB10	Implementation	Analyst, Developer, User
TB10	Critical Appraisal & Evaluation	Analyst
TB10	Lessons Learned	Analyst, Designer, Developer, Tester
TB10	Conclusion	Analyst

Fig 5.3: Time Boxing Chart

5.1.4 Gantt Chart

A Gantt Chart is a visual representation of the project activity schedule. It displays the timeline of tasks as progress bars, indicating their duration from the start to the end date, rather than breaking them down by days. The Gantt Chart for the *DeshiTalent – Bangladeshi Freelancer Hub (DTBFH)* project is provided below:

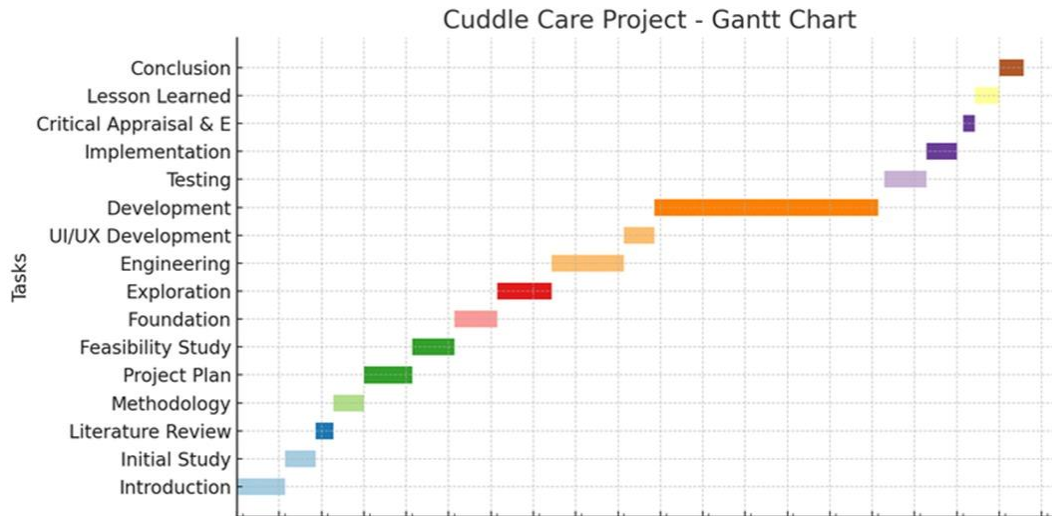


Fig 5.4: Gantt Chart

5.2 Test Plan

A comprehensive software test plan serves as a crucial document that defines the scope, objectives, and implementation of testing activities. It provides a structured framework to ensure the quality and reliability of the software by outlining the testing process in detail. It includes a strategic plan that outlines objectives, a detailed timeframe, and a comprehensive list of necessary resources, such as staff, software, and technology.

The test plan assumes paramount importance in the overall software testing framework, serving as the linchpin for ensuring the systematic and sequential implementation of all planned activities. Conceptually, the test plan functions as a standardized template for conducting software testing operations, thereby establishing a well-defined and rigorously controlled process under the supervision of the testing manager.

5.2.1 Testing Against the Time Boxes

The time box procedure involves predicting a defined maximum duration for a certain section. Evaluating within time constraints

Username	Example	Role	Example
Time box ID			
Time box content			

Table 5.1: Testing Against the Time Boxes

Different type of test and their test cases, expected result, actual results and comments table are following:

Test Type	Test cases	Expected Result	Actual Results	Comment
Unit Test				
Integration Test				
System Test				
Acceptance Test				
Security Test				
Usability Test				
Reliability Test				

Table 5.2: Testing and Test plan activity

5.2.2 Required Test

Functional and Non-Functional testing objective and scope and Using tools are following table:

Test Name	Objective	Scope	Tools
Unit Testing	Validate individual components or units of code to ensure they perform as expected.	Smallest unit of functionality.	PyTest (Python).
Integration Testing	Test the interaction between multiple components or modules.	Focuses on ensuring that the components or modules work together as intended.	Postman (for API testing)
System Testing	Test the entire system to ensure all components function together properly.	Entire system in an integrated environment (end-to-end testing).	QTP (QuickTest Professional)
Smoke Testing	Run a preliminary test to determine if the basic functionalities of the system are working.	A shallow, broad test to check the stability of the build before more detailed testing.	Selenium
Regression Testing	Make sure that any bug fixes or new modifications don't interfere with already-existing functionality.	Re-tests the application to verify that previously working features are still functional after updates.	Selenium
User Acceptance Testing (UAT)	Validate the application from the end user's perspective to ensure it meets business requirements.	Testing is typically done by actual users or clients to confirm that the system fulfills their needs and requirements.	Manual testing, but Selenium for automation.

Table 5.3: Required Test of Proposed Project

Test Case

Test Case No			
Test Type			
Test Description			
Test Steps	Expected Result	Actual Result	Comment

Table 5.4: Test Cases form

User Acceptance Test Plan

Test Case No			
Test Type			
Test Description			
Precondition of testing			
User's name			
Act as			
Test Steps	Expected Result	Actual Result	Comment

Table 5.5: User acceptance test plan form

5.3 Risk Management

Risk management is the process of identifying, analyzing, and addressing potential risks that may arise during the project development lifecycle. This approach helps in monitoring the project's progress and ensures the achievement of its goals and desired outcomes. Risks associated with a project may involve its timeline, budget, or overall performance. The key components of risk management include:

- ✓ **Risk Identification**
- ✓ **Risk Assessment**
- ✓ **Risk Mitigation/Action Plan**
- ✓ **Steps to Address Potential Risks**

Risk Identification

Risk identification is a critical component of risk management planning. It involves the initial step of recognizing potential risks that could impact the project's progress, timeline, budget, or performance. This process is essential to proactively address challenges and minimize their effects on the project are following:

- ✓ **Market Adoption Risk:** Target audience may resist switching from existing global platforms or lack awareness of local platform benefits.
- ✓ **Payment System Integration Risks:** Challenges in integrating local payment gateways (bKash, Nagad) or third-party payment providers.
- ✓ **Technical Risks:** Unforeseen technical issues during development, such as performance bottlenecks, security vulnerabilities, or bugs.
- ✓ **Legal and Regulatory Compliance:** Need to comply with local laws related to data protection, payments, freelancing regulations, and tax policies.
- ✓ **Competition Risk:** Established global platforms may increase their focus on Bangladesh, creating tougher competition.
- ✓ **Scalability and Infrastructure Risks:** Platform may struggle to scale if user traffic surges, especially if infrastructure is not designed for large-scale growth.
- ✓ **User Trust and Reputation Risks:** Fraudulent activity, disputes, or poor-quality work may damage the platform's reputation.
- ✓ **Cultural and Language Barriers:** Potential misunderstandings between users due to differences in cultural expectations, communication style, or language preferences.
- ✓ **Payment Fraud and Disputes:** Clients or freelancers may engage in fraudulent activities, such as non-payment or low-quality work.
- ✓ **Changing Market Trends:** The freelancing landscape may evolve with new tools, technologies, or platforms, potentially affecting Desh Talent's relevance.
- ✓ **Cybersecurity Risks:** The platform could become a target for hacking or data breaches.
- ✓ **User Retention and Engagement Risks:** Users may abandon the platform if it does not offer sufficient incentives, job opportunities, or a seamless experience.

5.4 Change Management Approach

The DeshiTalent - Bangladeshi Freelancer Hub project will adhere to a rigorous change management framework to effectively navigate and adapt to evolving project circumstances. This framework will encompass the following key elements:

- ✓ **Identification of Potential Change Drivers:** The project will proactively identify and monitor potential change-inducing factors, such as shifting market dynamics, advancements in technology, user feedback, and unforeseen technical hurdles. This proactive monitoring will enable the project team to anticipate and prepare for potential adjustments.
- ✓ **Embracing Change with Adaptability:** The project will employ an agile methodology, fostering flexibility to incorporate necessary changes that align with evolving business objectives or enhance user experience. This agile approach will prioritize adaptability while ensuring that changes do not compromise project timelines, budgets, or overall objectives.
- ✓ **Formal Change Request Procedure:** All proposed changes will undergo a formal evaluation process involving the project manager and key stakeholders. This assessment will meticulously evaluate the potential impact of the proposed change on project scope, timeline, resource allocation, and budget. Following this comprehensive evaluation, a decision will be made to either approve or reject the proposed change.
- ✓ **Transparent Communication:** Effective communication regarding any changes will be maintained with all relevant team members, clients, and stakeholders. This transparent communication ensures that all parties remain informed and aligned throughout all phases of the project.
- ✓ **Post-Change Assessment:** Upon the implementation of any changes, a thorough post-change evaluation will be conducted. This evaluation will rigorously assess the effectiveness of the implemented changes in achieving their intended outcomes and identify any potential risks or issues that may have arisen because of the changes.

Considering the business priority

Changes were considered to achieve the company's intended goals and objectives. Therefore, it is necessary to prioritize features and needs alignment that will greatly benefit the consumer.

Topical Area	Priority Level
Delivering	5
Functional requirements	8
Non - Functional requirements	8
Segments for rating and reviews	7
Segments for security handling	10

Table 5.6: Considering business priorities

5.5 Standard Quality Measures

Ensuring quality in both the solution and the development process is a critical aspect of delivering a successful project. The measures taken to achieve quality standards are outlined below:

Solution Quality

The quality of the solution has been assured by aligning it with the expected business needs and user requirements. This was achieved by systematically tracking user expectations and prioritizing tasks. Two key prioritization techniques were utilized to ensure the timely and effective delivery of the solution:

- ✓ **Time Boxing:** Dividing tasks into manageable timeframes to ensure timely completion.
- ✓ **MoSCoW Method:** Prioritizing requirements into Must-Haves, Should-Haves, Could-Haves, and Won't-Haves to focus on the most critical needs.

These approaches were carefully implemented while adhering to the project schedule.

Process Quality

The quality of the development process has been maintained by adhering to the standards relevant to the target organization. Two methodologies were employed to ensure process quality:

- **Capability Maturity Model Integration (CMMI):** A framework to improve process performance and achieve measurable results.
- **Dynamic Systems Development Method (DSDM):** An Agile approach to delivering high-quality solutions within fixed timeframes and budgets.

These measures collectively ensured that the project met the desired quality standards while adhering to the planned schedule and objectives

Chapter 6 - Feasibility Analysis

6.1 Possible feasibility analysis

A **feasibility analysis** is essential to assess whether **DeshiTalent - Bangladeshi Freelancer Hub** can be successfully developed, implemented, and sustained. This analysis helps evaluate the project's operational, technical, financial, and market feasibility. Below is a breakdown of each type of feasibility analysis:

Operational feasibility

Operational feasibility focuses on the usability, ease of use, and required functionalities of the proposed system. The *DeshiTalent – Bangladeshi Freelancer Hub* has been designed with robust validation and verification processes to ensure accurate user inputs. It facilitates seamless communication between clients and freelancers, enhancing overall efficiency. The project also considers the availability of key professionals in Bangladesh, including software developers (frontend and backend), UI/UX designers, and QA engineers, to ensure smooth development and implementation. Additionally, to secure payment transactions, the system integrates with local payment gateways. This ensures streamlined transaction management, reducing delays or obstacles in fund transfers and providing users with a reliable and efficient payment experience.

Technical Feasibility

A technical feasibility assessment of "DeshiTalent – Bangladeshi Freelancer Hub" indicates that users can efficiently manage their tasks through the system, representing a significant improvement over previous manual method. Development to using current and widely adopted web and mobile technologies, the system is designed for broad compatibility, functioning effectively across various web browsers even with limited internet connectivity. Secure data access is ensured through a hierarchical access control system. Furthermore, the web-based application's platform independence, coupled with the affordability and cost-effectiveness of the chosen development platform and resources, confirms its technical feasibility.

Category	Component
Hardware	HP laptop (Configuration) Wi-fi Router
Software	Microsoft office Google Chrome Browser Windows 11 (Operating system) Visual Studio
Database	SQLite3
Technology	<u>Client Side:</u> I. Html II. CSS III. JavaScript IV. Tailwind <u>Server Side:</u> I. Python (Version - 3.10) II. Django (Version - 5.1)

Table 6.1: Technology to used of proposed DeshiTalent System

Web based application cost:

The entire application and its data are hosted on a remote server, allowing users to access it anytime and from anywhere via the Internet. This eliminates the need for any installation, ensuring convenience and ease of use.

Equipment	Cost per Unit	Total Cost
Extranet network with VPN	₹ 3000 per month	₹ 3,000
Desktop PC (Core i9, 3.6 GHz processor, 32 GB DDR4 RAM, 2 TB HDD)	₹ 1,50,000	₹ 1,50,000
Web, File, and Email servers	₹ 15,000 per month	₹ 15,000
Total		₹ 1,68,000

Table 6.2: System Configuration

Cost Benefit Analysis:

SL No.	Equipment	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Web based application cost	₹1,68,000	-	-	-	-	₹1,12,000
2	Domain and hosting Cost	₹ 25,000	₹25,000	₹25,000	₹25,000	₹25,000	₹ 1,25,000
3	Employee Expenses	₹ 40,000	₹40,000	₹40,000	₹40,000	₹40,000	₹2,00,000
4	Other Cost	₹ 25,000	₹25,000	₹25,000	₹25,000	₹25,000	₹1,25,000
	Total Cost	₹2,58,000	₹90,000	₹90,000	₹90,000	₹90,000	₹6,18,000

Fig 6.1: Cost Benefit Analysis of proposed system

Chapter 7 - Foundation

7.1 Problem Area Identification

Identifying the problem is key to improving any project. In this case, it is essential to gather input directly from the users, as they will be the primary individuals interacting with the system. Based on user feedback, several problem areas have been identified, as outlined below:

7.1.1 Interview

To effectively gather information and understand the specific issues or desired outcomes, conducting an interview with users is the most effective method. Identifying the root cause of problems is crucial for formulating the best solution. As such, a set of targeted questions has been developed to ask users during the interview process, with the aim of obtaining valuable insights to inform the development of the proposed system.

1. For Freelancer:

- a. Is there any possible of Access to local opportunities
- b. What payment methods are available for freelancers, and how is payment security ensured?
- c. Are there any rating or review systems for freelancers? If so, how are they managed?
- d. How does the bidding or job application process work for freelancers?

2. For Client:

- a. How does the platform ensure that freelancers meet the client's requirements?
- b. What happens if a client is unsatisfied with a freelancer's work?
- c. How does the payment system work from the client's side? Are there any escrow mechanisms?
- d. How is data privacy and security maintained for users?

3. Admin:

- a. What are the main roles and responsibilities of the admin in DeshiTalent?
- b. How does the admin manage user verification (both freelancers and clients)?
- c. What steps are taken to prevent fraudulent activities on the platform?
- d. How does the admin monitor and resolve disputes between freelancers and clients?
- e. How does the platform ensure quality and trust between freelancers and clients?

7.1.2 Observation

Observational techniques are highly valuable for gathering user requirements and identifying the root of the problem. Primarily, these techniques involve immersing oneself in the user's workplace to observe their behavior and identify solutions that are more likely to be accepted by them. Key considerations for effective observation include:

For Freelancers:

- How easy is it for freelancers to create and verify their profiles on the platform?
- Are freelancers able to find jobs relevant to their skills and experience?
- Is the competition among freelancers on the platform fair and balanced?
- Do freelancers encounter difficulties in understanding client requirements?

For Client:

- Is the project posting process straightforward and user-friendly?
- Are clients satisfied with the escrow system and payment milestones?
- Do clients feel confident in the platform's verification process for freelancers?

General Observations:

- Are users (freelancers, clients, admins) finding the platform intuitive and easy to navigate?
- How well does the platform attract and retain active users?
- How well does the platform handle high traffic or concurrent usage?

7.2 Implemented technology

To implement the suggested system, a number of implementation technology solutions are available. Selecting the appropriate technology is crucial for achieving success. The options are as follows:

Server Site Application Technology

The client-server model constitutes a distributed application architecture characterized by a division of labor between service providers (servers) and service consumers (clients). In this paradigm, client computers initiate data requests to servers via the internet. Upon receiving a request, the server processes it and subsequently transmits the requested data packets back to the originating client. A salient feature of this architecture is the absence of resource sharing among clients. (GeekforGeeks)

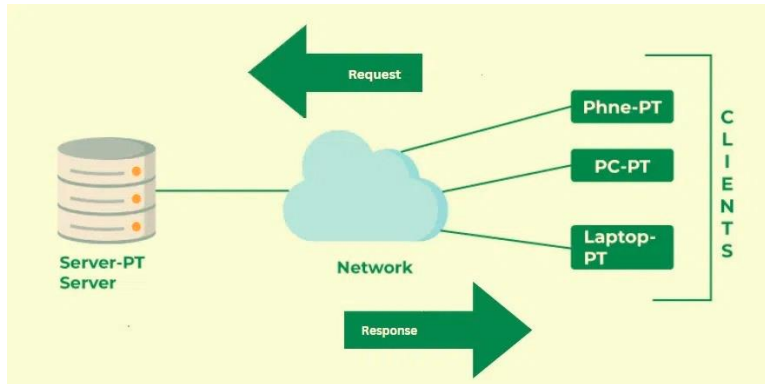


Fig 7.1: Server – Client-Side Technology

How It Works

1. Request and Response:

- ✓ A client sends a request to a server using the network.
- ✓ The server processes the request and sends back the required data or response.

2. One-to-Many Communication:

- ✓ A single server can handle requests from multiple clients simultaneously.

3. Stateless or Stateful:

- ✓ Stateless: Each request is treated independently by HTTP protocol.
- ✓ Stateful: The server retains the client's session.

Key Features

- ✓ **Centralized Resources:** Servers host and manage data, applications, or services centrally, ensuring consistency and control.
- ✓ **Scalability:** Servers can be scaled to handle increasing client requests
- ✓ **Separation of Concerns:** Clients concentrate on user engagement, whereas servers manage data processing and storage.
- ✓ **Security:** Servers implement centralized security measures like authentication and encryption.

Advantages

- ✓ **Efficient Resource Management:** Servers centralize resources, making management and updates easier.
- ✓ **Data Integrity:** Centralized data ensures consistency, as all clients interact with the same data source.
- ✓ **Accessibility:** Clients can access the server's resources from anywhere with a network connection.
- ✓ **Scalability:** Additional servers can be added to handle increased client demands.

7.8 Recommendation and Justification

To ensure global accessibility and accommodate a diverse user base, the proposed *DeshiTalent – Bangladesh Freelancer Hub* necessitates clients prioritize user engagement, whereas servers manage data processing and storage. A system available 24/7 from any internet-enabled device, including cellphones, laptops, desktops, and tablets. Furthermore, the system must be capable of concurrently supporting a substantial number of users. Given these requirements, a traditional client-server application architecture proves inadequate. Consequently, a web application is recommended as the most suitable solution for the proposed system

Chapter 8 - Exploration

8.1 Full Use of System

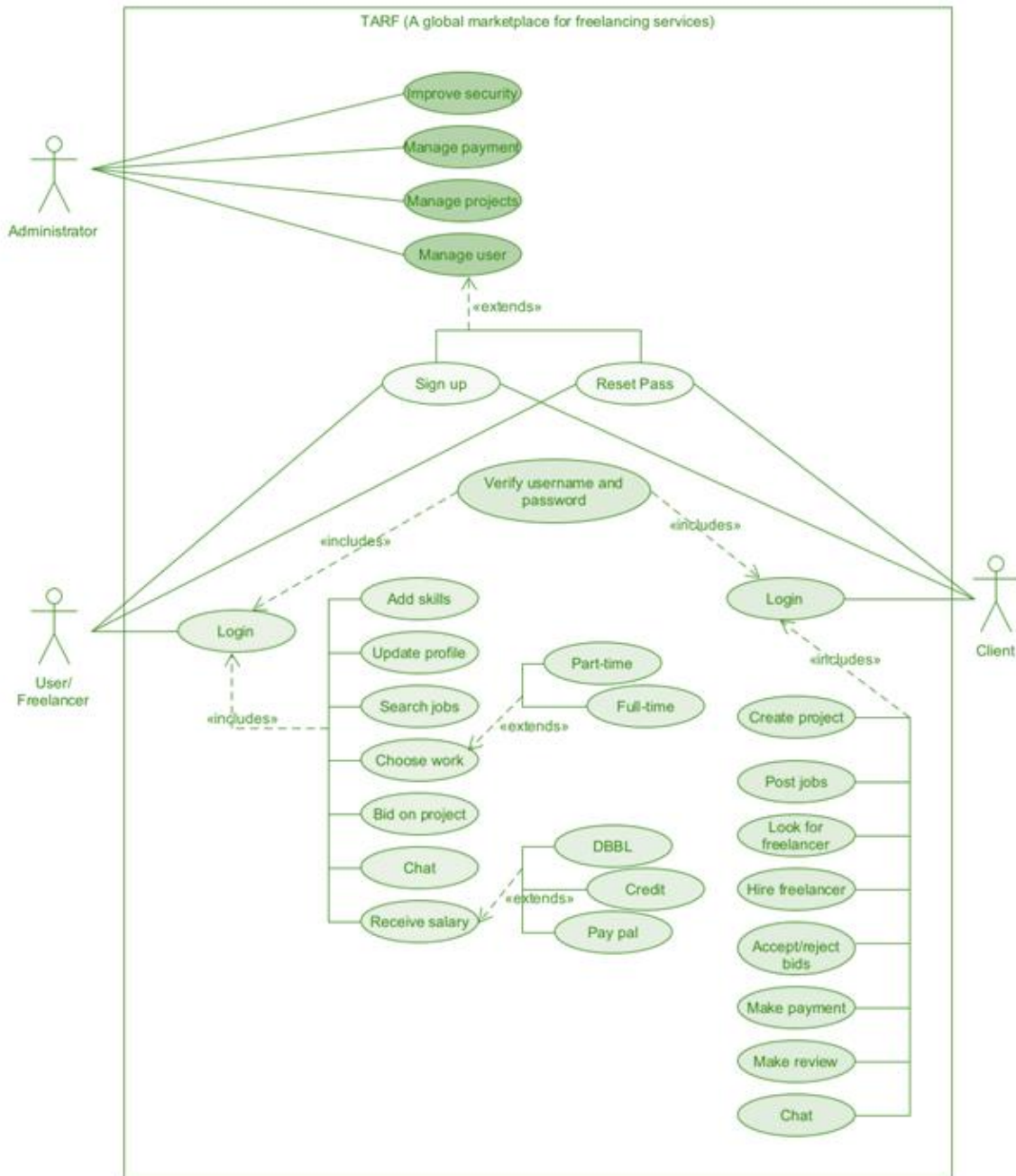


Fig 8.1: Use case Diagram of DeshiTalent

8.2 Activity Diagram of System

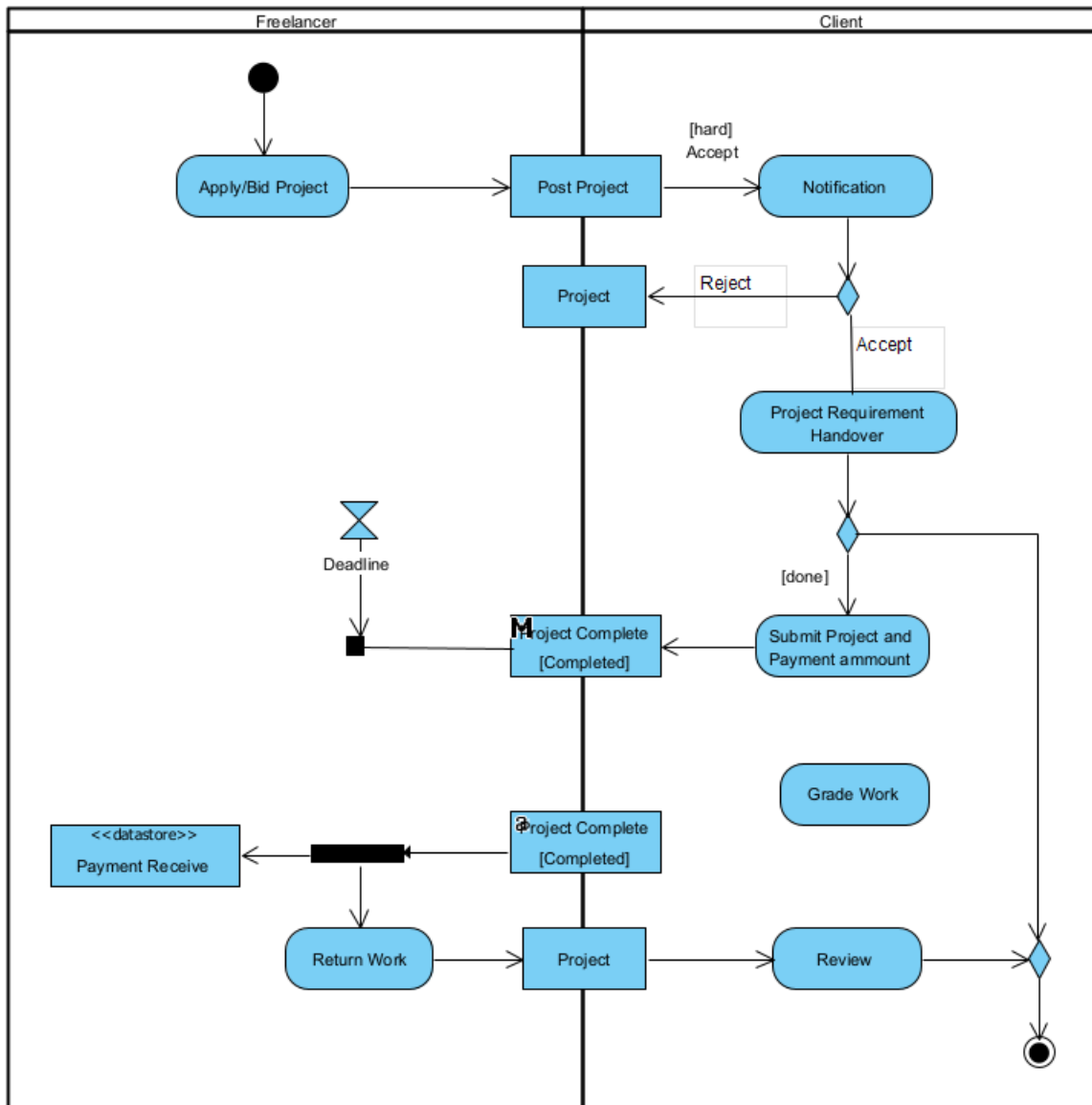


Fig 8.2: Activity Diagram of DeshiTalent

8.3 Requirements Catalogue

Requirements Catalogue for DeshiTalent – Bangladeshi Freelancer Hub

ID	Requirement Name	Type	Description	Priority	Dependencies	Stakeholders
FR001	User Authentication	Functional	Users can register, log in, and log out with role-based access (Freelancer, Client, Admin).	Must-Have	None	Freelancers, Clients, Admin
FR002	Role Management	Functional	Different roles have specific permissions and access levels.	Must-Have	User Authentication	Admin
FR003	Project Posting	Functional	Clients can post projects with details like title, description, budget, and deadlines.	Must-Have	Role Management	Clients
FR004	Bidding System	Functional	Freelancers can view and bid on projects with proposed costs and timelines.	Must-Have	Project Posting	Freelancers
FR005	Payment Gateway Integration	Functional	Secure payment transactions through SSLCommerz with an escrow system.	Must-Have	Project Completion, Role Management	Freelancers, Clients
FR006	Messaging System	Functional	Real-time messaging between freelancers and clients for project discussions.	Must-Have	Project Posting, Role Management	Freelancers, Clients

FR007	Dashboard	Functional	Personalized dashboards for tracking projects, payments, and progress.	Must-Have	Role Management	Freelancers, Clients, Admin
FR008	Admin Panel	Functional	Tools for managing users, projects, payments, and disputes.	Must-Have	Role Management	Admin
FR009	Rating and Review System	Functional	Users can rate and review others after project completion.	Must-Have	Project Completion	Freelancers, Clients
NFR008	Portability	Non-Functional	The system should be deployable across different hosting environments with minimal changes.	Could-Have	Deployment Strategy	Developers

Table 8.1: Requirements Catalogue of DeshiTalent Project

8.4 Prioritized Requirements List (PRL)

Prioritized Requirements List (PRL) for DeshiTalent - Bangladeshi Freelancer Hub:

Requirement	Description	Priority
User Authentication and Role Management	Secure registration, login, and role-based access for freelancers, clients, and admins.	Must-Have
Project Posting	Allow clients to post projects with detailed descriptions, budgets, and deadlines.	Must-Have
Bidding System	Enable freelancers to submit bids on projects with proposed timelines and costs.	Must-Have

Payment System Integration (SSLCommerz)	Implement secure escrow-based transactions for payments between clients and freelancers.	Must-Have
Messaging System	Provide a real-time chat system for freelancers and clients to communicate effectively.	Must-Have
Dashboard for Freelancers and Clients	Personal dashboards to track projects, payments, and activity history.	Must-Have
Admin Panel	Tools for platform management, user monitoring, dispute handling, and analytics.	Must-Have
Rating and Review System	Allow users to rate and review after project completion for building trust and transparency.	Must-Have
Notification System	Notify users about important updates such as project bids, messages, and payment status.	Should-Have
Reporting and Analytics	Provide insights and analytics to track performance, earnings, and trends for all users.	Should-Have
Skill Verification and Certification	Enable freelancers to verify their skills through certifications or tests.	Could-Have
Portfolio Showcase	Allow freelancers to display their portfolio of completed projects.	Could-Have
Project Milestone Tracking	Enable tracking of project progress through milestones for better management.	Could-Have

Table 8.2: Prioritized Requirements List of DeshiTalent

8.4 Prototype of the new system

Login Pages

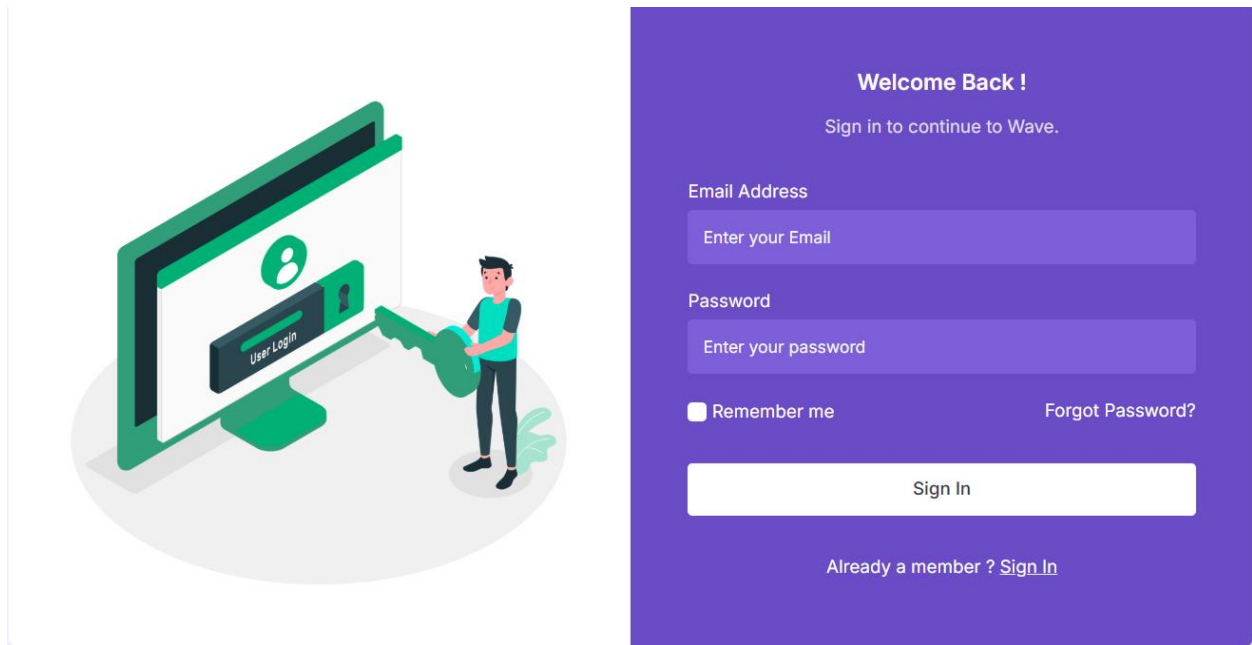


Fig 8.3: Login Pages

Signup

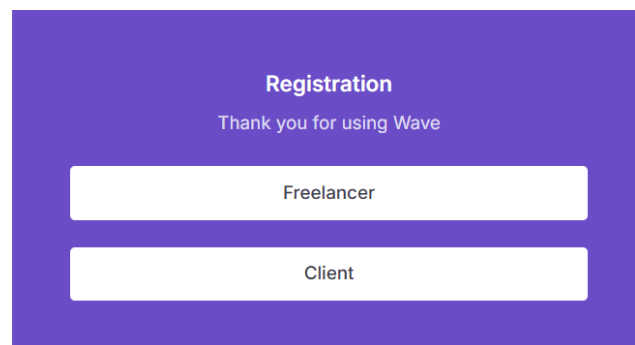


Fig 8.4: Signup Pages Part-1

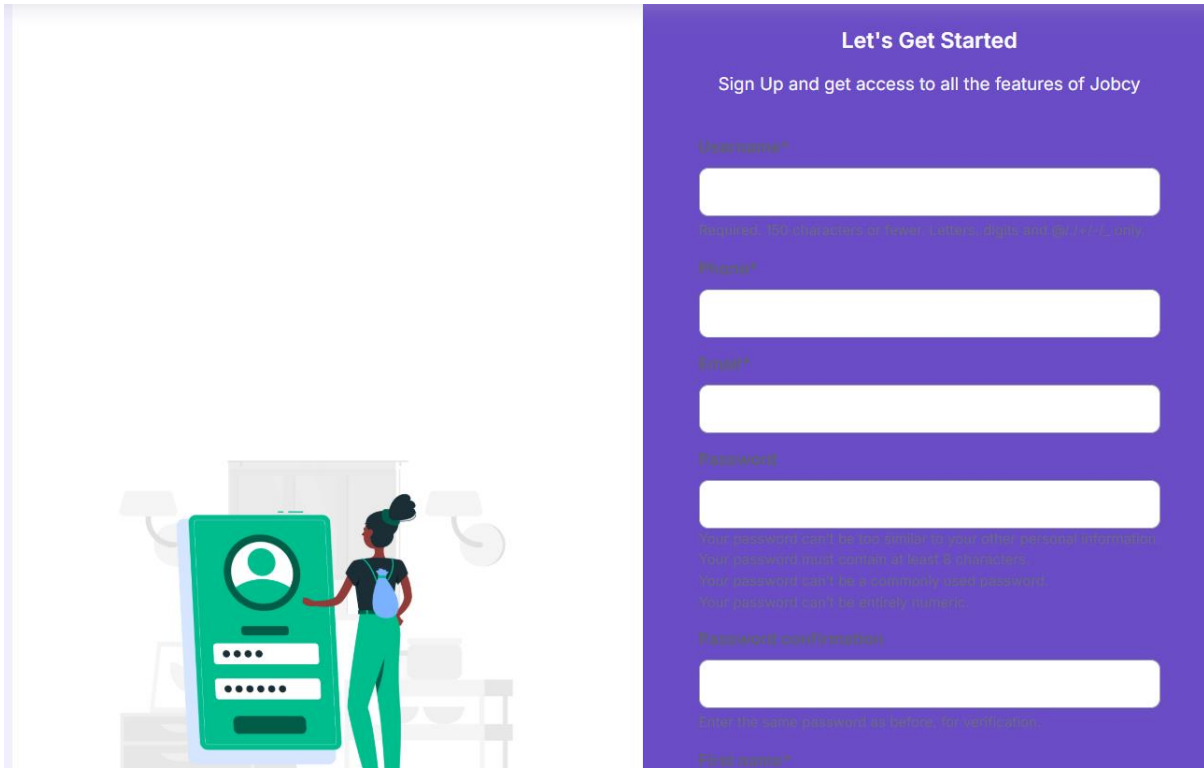


Fig 8.5: Signup Pages Part-II

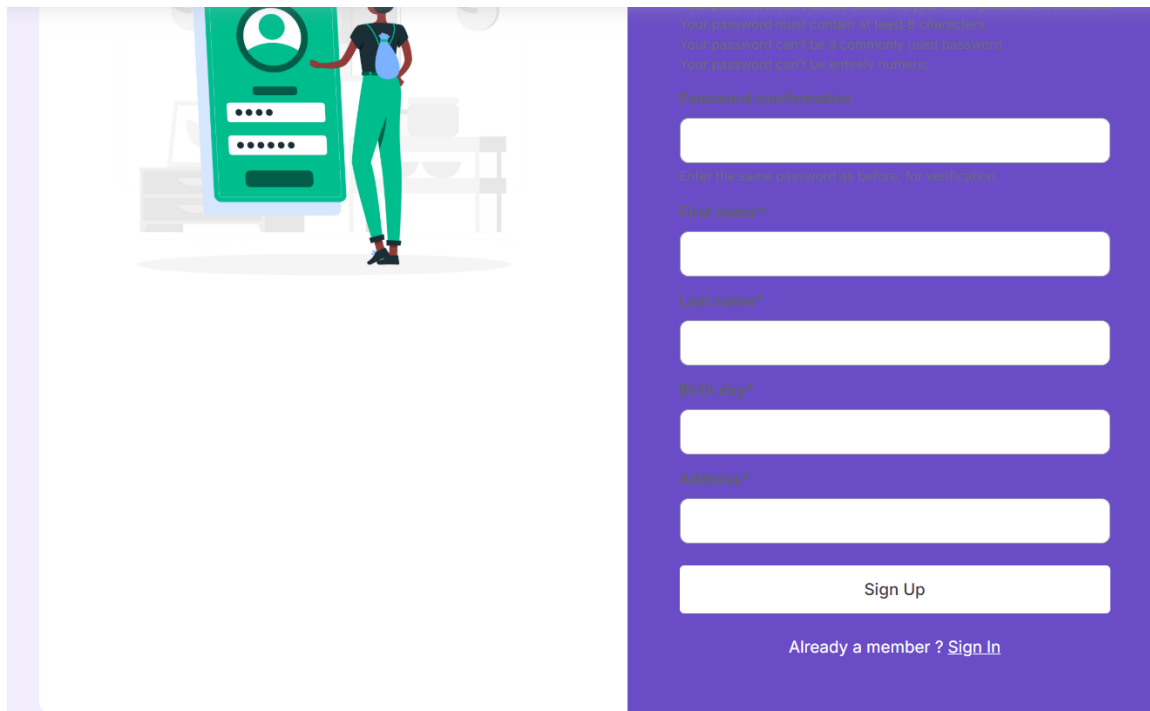


Fig 8.6: Signup Pages Part-III

Authentication

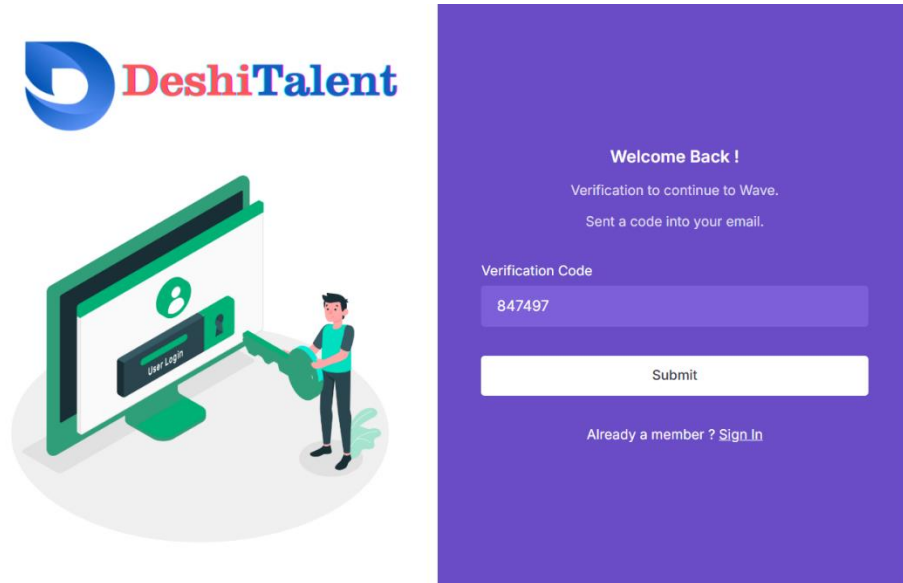


Fig 8.7: Authentication Page

Dashboard of Freelancer

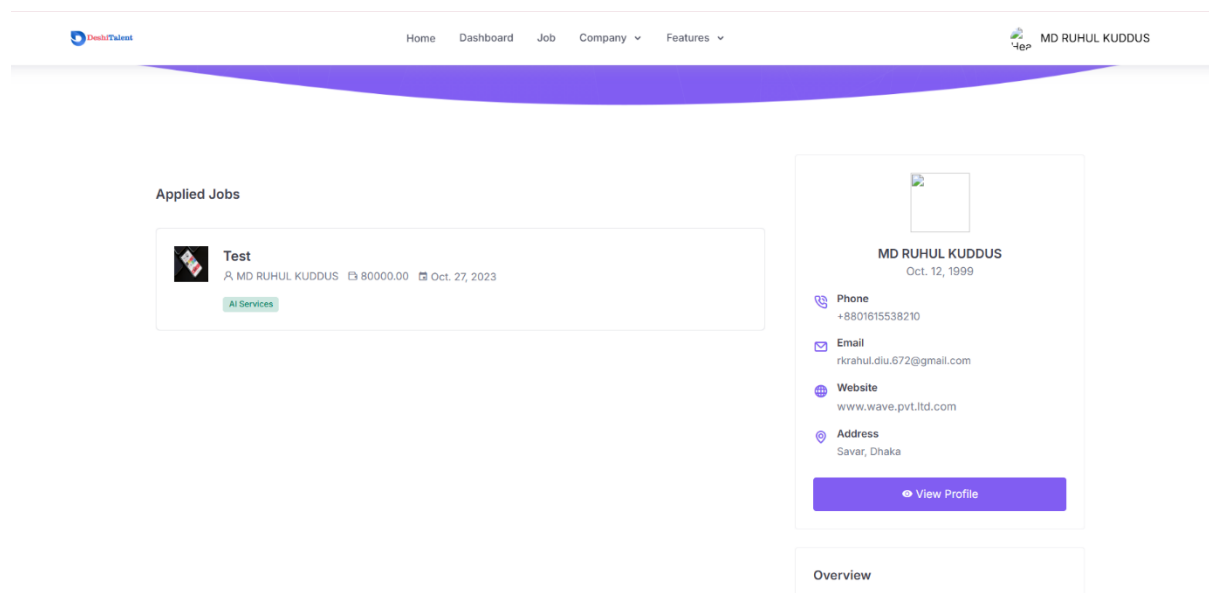


Fig 8.8: Dashboard of Freelancer

Project List

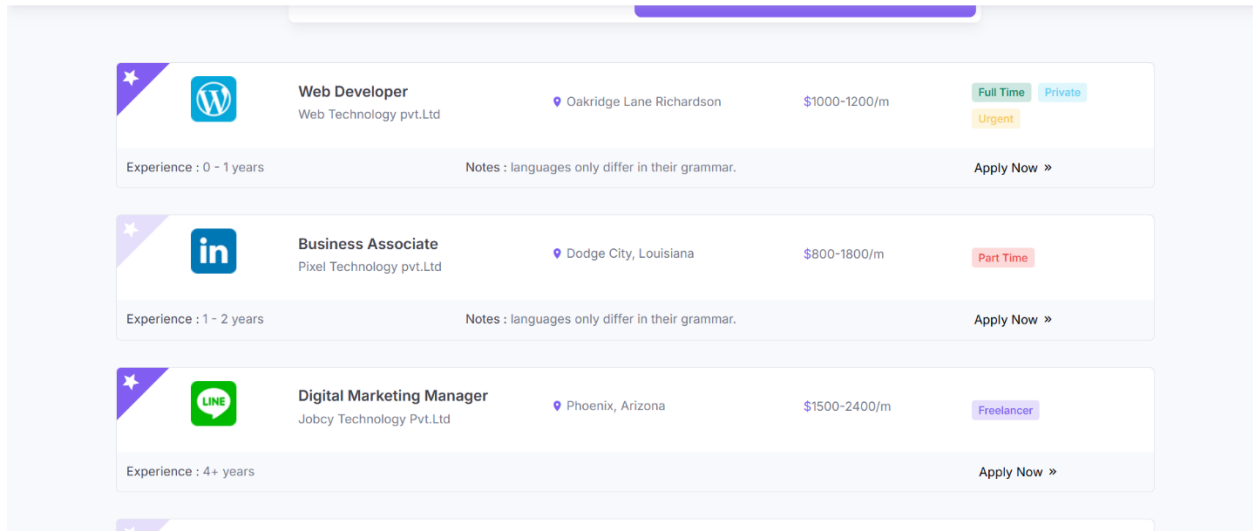


Fig 8.9: Show Project List

Bookmark

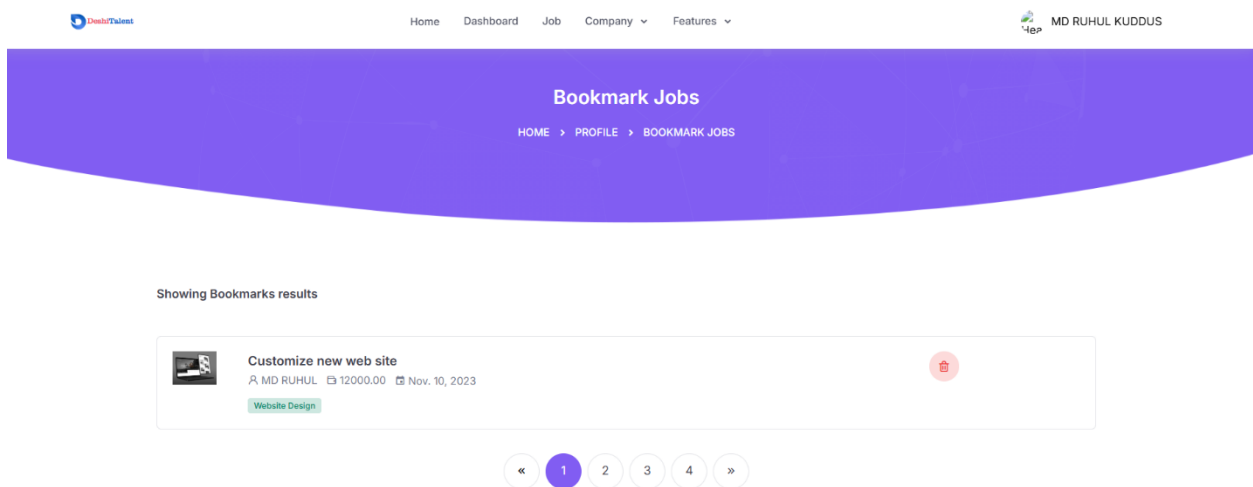


Fig 8.10: Bookmark pages

Registration Process

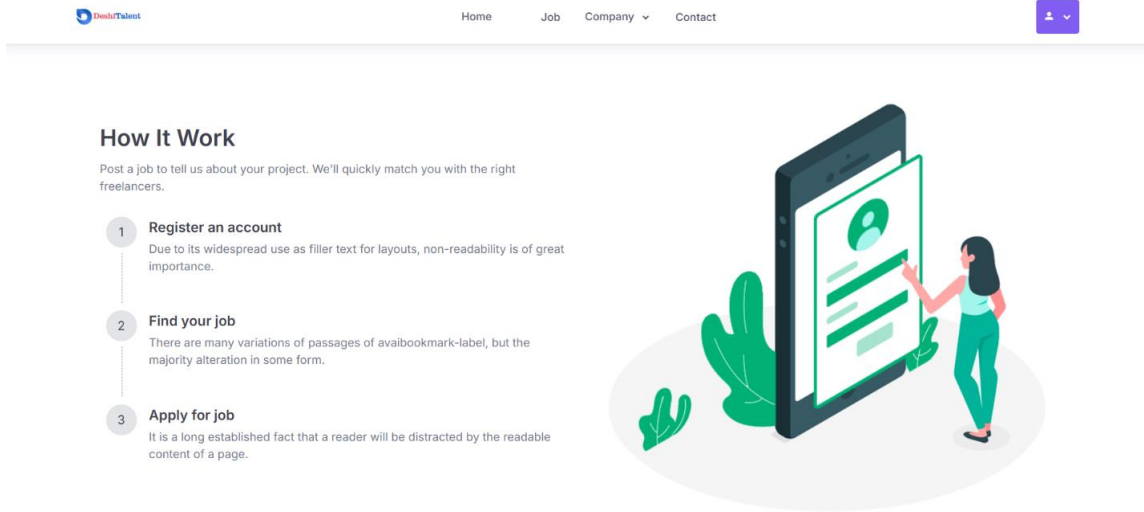


Fig 8.11: Show the registration Process

Project Posting

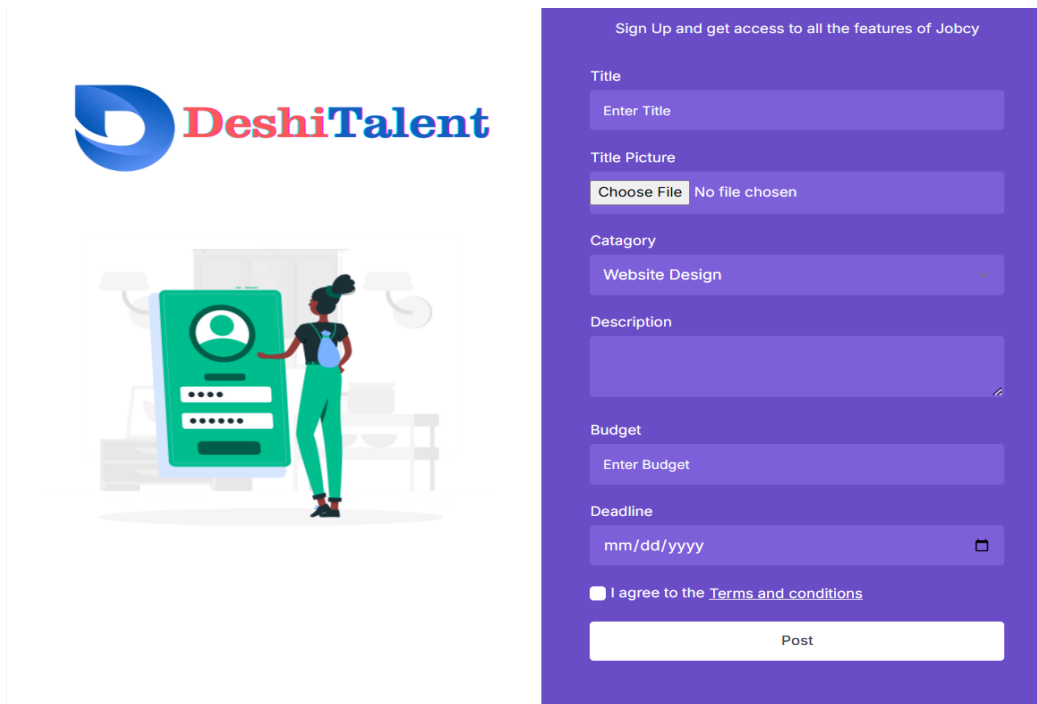


Fig 8.12: Client Post the Project

Project Details

DASH/Talent Home Dashboard Job Company Features

Test
MD RUHUL KUDDUS 4.8 ★★★★★

Deadline
Oct. 27, 2023

Job type
AI Services

Budget
\$ 80000.00

Job Overview

- Job Title**
Test
- Deadline**
Oct. 27, 2023
- Offered Salary**
\$ 80000.00
- Date Posted**
Posted 2 hrs ago

[Apply Now](#)

[Add Bookmark](#)

Fig 8.13: Project Details show

Chapter 9 - Engineering

9.1 Use Case Diagram

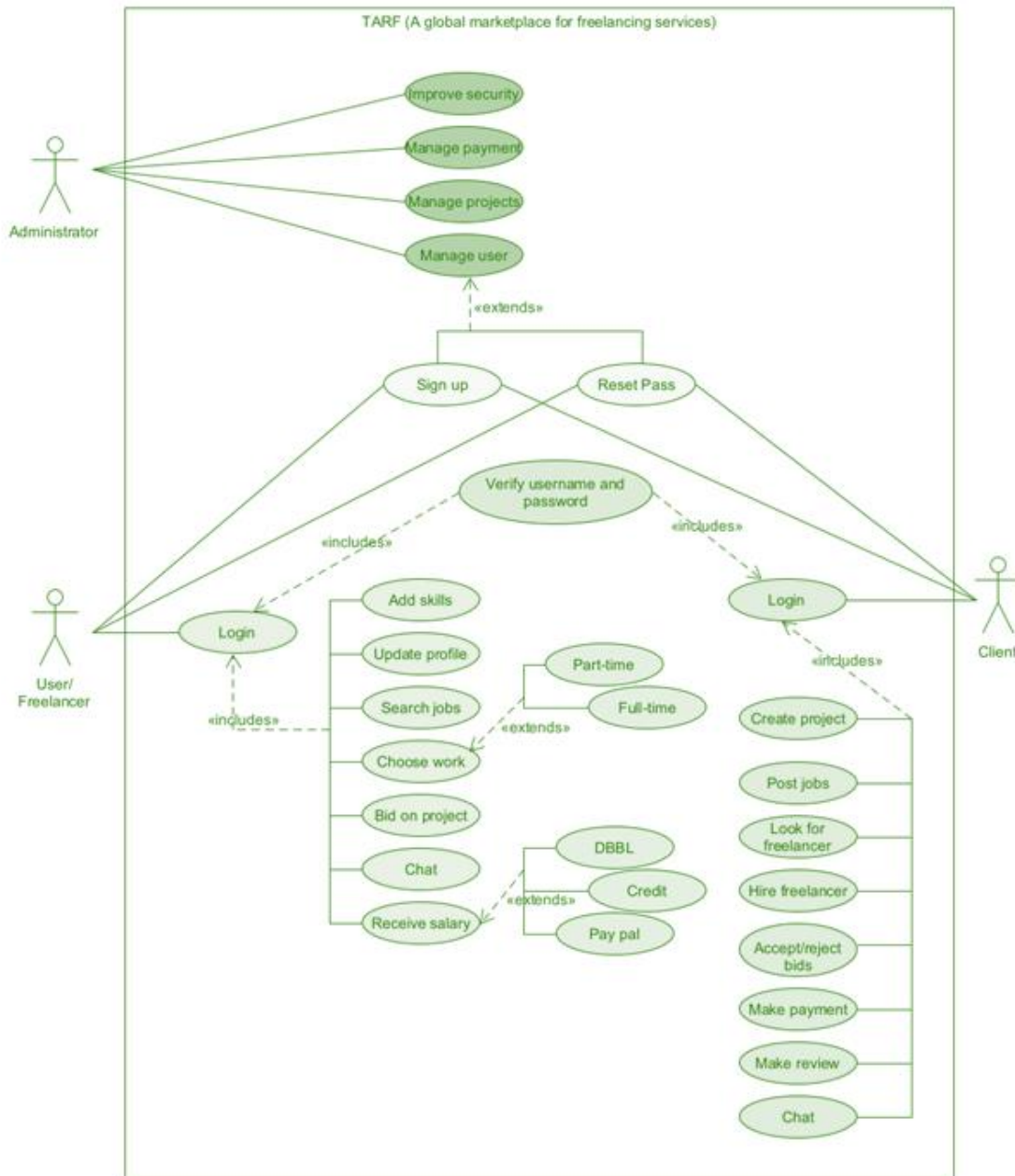


Fig 9.1: Use Case Diagram of DeshiTalent

9.2 ER diagram of DTBFH

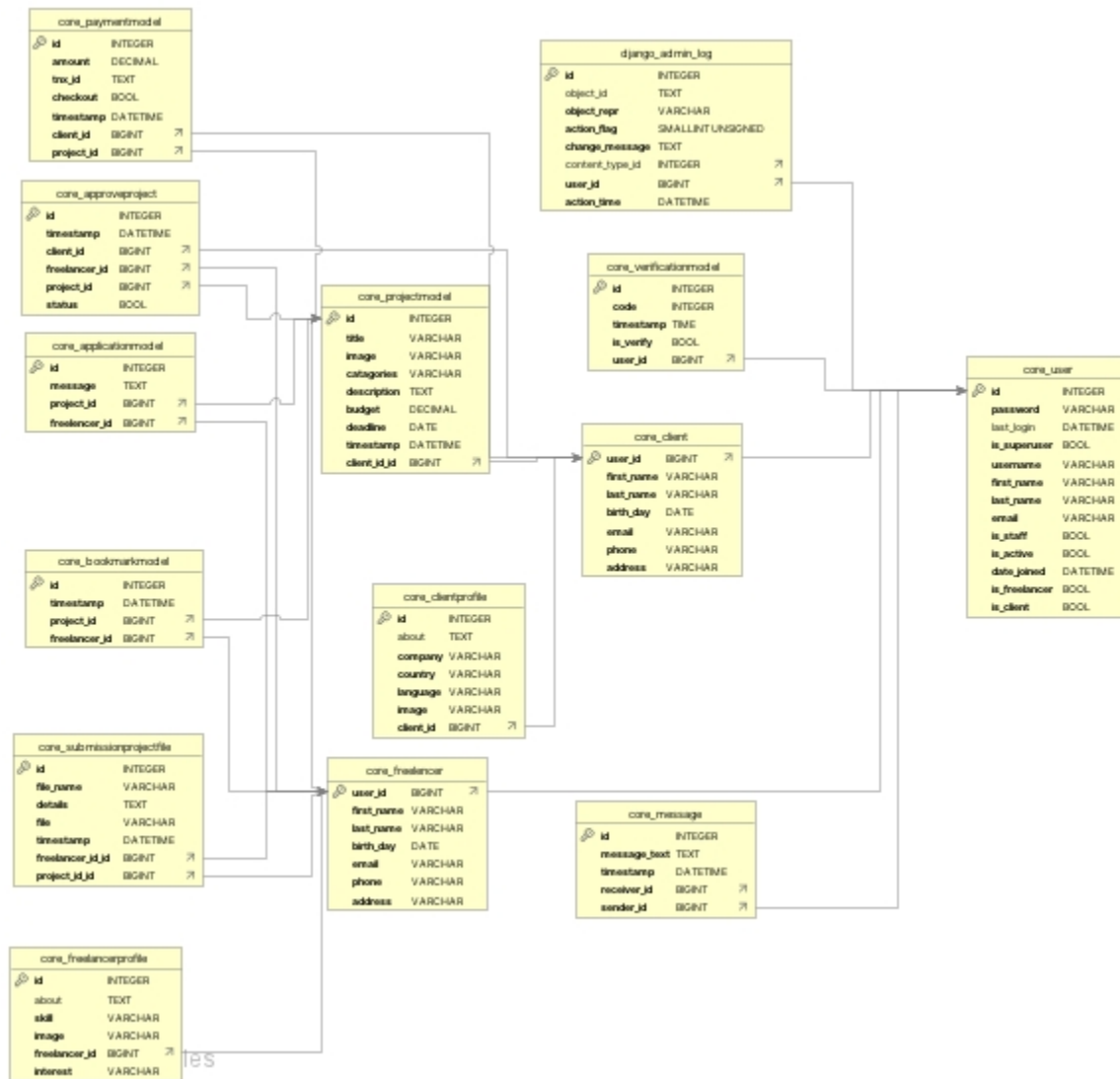


Fig 9.2: ER diagram of DeshiTalent

9.2 Sequence Diagram of DTBFH

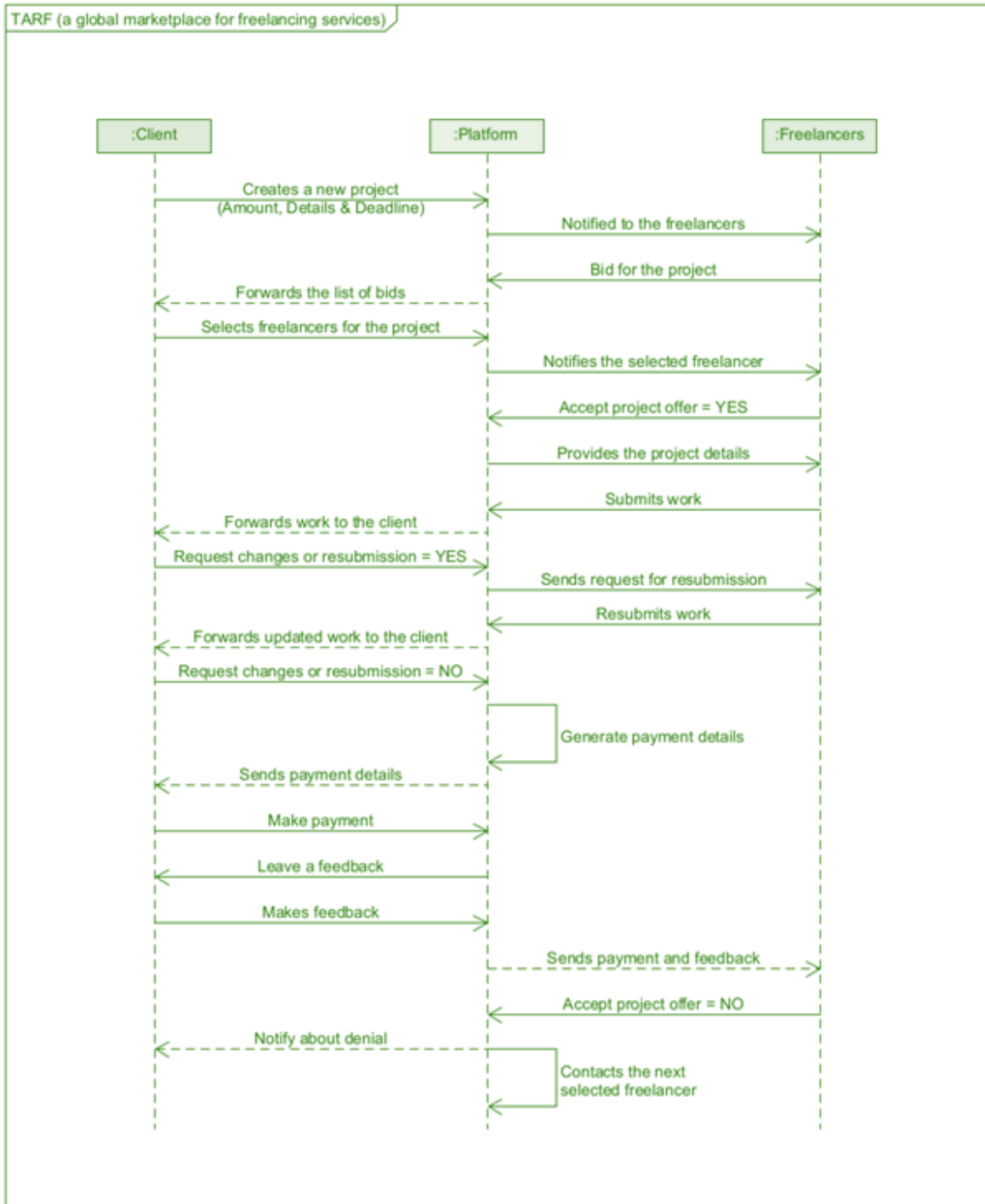


Fig 9.3: Activity Diagram of DeshiTalent

9.3 Full System Activity Diagram

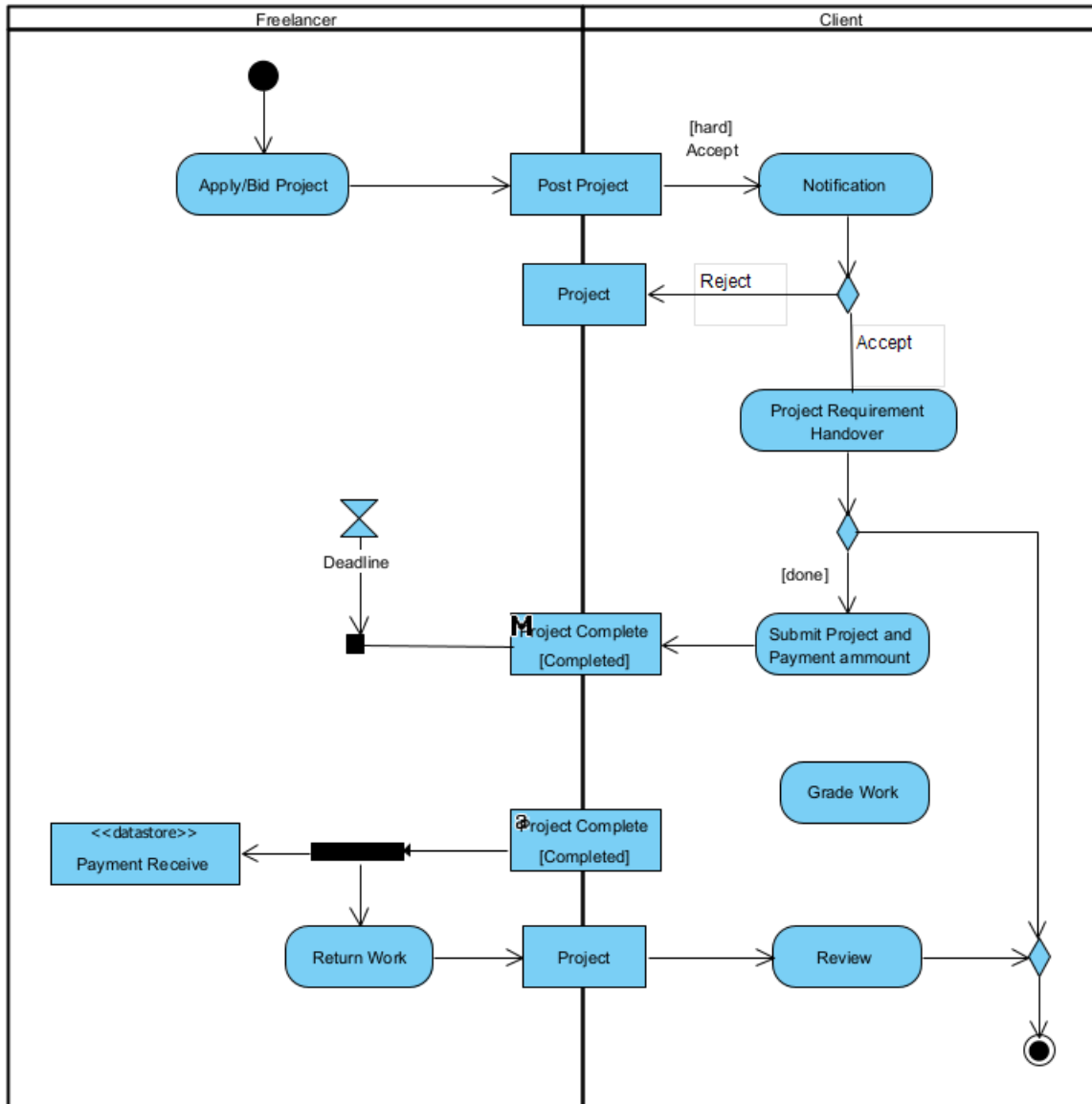


Fig 9.4: Full System Activity of DeshiTalent

9.4 Logical Diagram of MVT Architecture

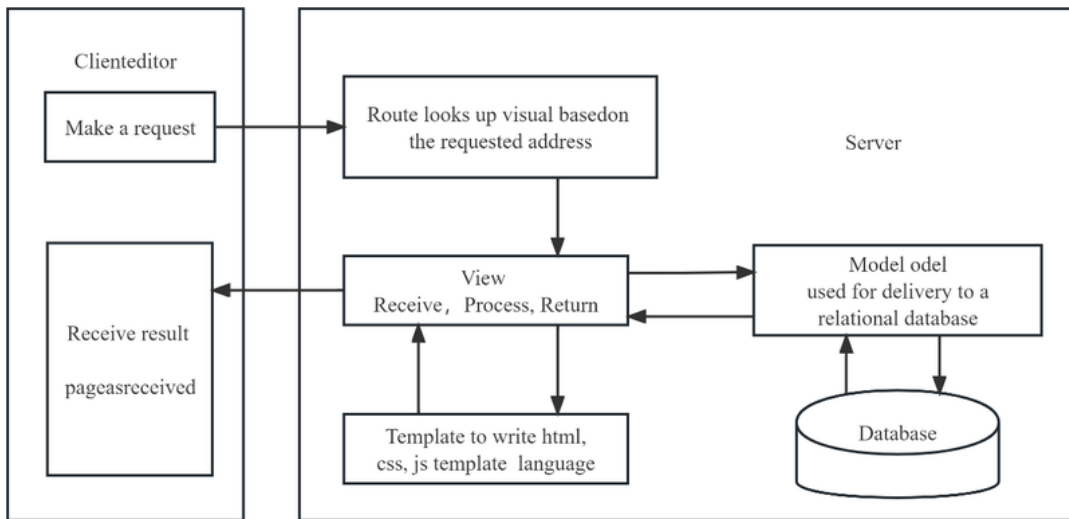


Fig 9.5: Logical Diagram of DeshiTalent

Chapter 10 - Testing

10.1 Functional Testing

Functional testing is aimed at validating the core functionalities of the *DeshiTalent* platform to ensure it meets the specified requirements. This testing can be broken down into three key types:

- **Unit Testing:** Testing individual components or units of the system to verify that each functions correctly in isolation.
- **Module Testing:** Evaluating a group of related components or modules to ensure they work together as intended.
- **Integration Testing:** Verifying that different modules or systems interact correctly when integrated, ensuring seamless communication and data flow between them.

10.1.1 Unit Testing

Unit Testing validates individual components or functions to ensure that they are working as expected.

- ✓ **Form Field Validation for User Registration**
 - Test the form fields during user registration to ensure that all required fields (name, email, password, etc.) are validated correctly.
- ✓ **Freelancer Profile Update Validation**
 - Ensure that freelancers can update their profile information (skills, experience, etc.) and that validation is performed for each field (e.g., skill categories, price ranges).
- ✓ **Payment Calculation Functionality**
 - Verify the payment calculation functions, ensuring that the correct amount is calculated based on the project size and terms.
- ✓ **Job Posting Filtering (Client-side)**
 - Verify that job postings can be filtered by category, location, and budget, and that the list updates according to these filters.
- ✓ **Bidding and Proposal Filtering (Freelancer-side)**
 - Verify that bids can be filtered by price, skills required, and job categories, and that the list updates as per these filters.

10.1.2 Module Testing

Module Testing validates integrated functionalities or features that work together to complete specific tasks.

- **User Registration and Authentication**
 - ✓ Test the full registration process, including validation of input data, sending verification emails, and successfully logging in with valid credentials.
- **Freelancer Job Application Process**
 - ✓ Test the entire process where a freelancer searches for jobs, applies for a job by submitting a proposal, and waits for the client's acceptance.
- **Client Job Posting and Project Management**
 - ✓ Verify that the client can post a job, review freelance bids, accept/reject proposals, and modify or delete job listings as needed.
- **Payment System Integration**
 - ✓ Ensure that freelancers can receive payments through integrated payment systems (**bKash, SSLCOMMERZ**), and clients can pay securely.

10.1.3 Integration Testing

Integration Testing focuses on ensuring that different components of the platform work together as intended.

- **Login System with Valid Credentials**
 - ✓ Test that the user can log in using valid credentials and is directed to the appropriate dashboard based on their role (freelancer, client, or admin).
- **Job Application and Proposal Submission**
 - ✓ Ensure that the application and proposal system is integrated so that freelancers can apply to jobs, and clients can view and review proposals seamlessly.
- **Payment Processing**
 - ✓ Verify that once a project is completed, the system processes the payment correctly from the client's account to the freelancer's account, including handling escrow payments.
- **Notifications System**
 - ✓ Ensure that the system sends notifications to users (clients, freelancers, and admins) when key actions are taken (e.g., bid submission, job acceptance, payment confirmation).

10.2 Non-Functional Testing

Non-functional testing emphasizes evaluating elements of the platform that are not directly tied to certain functionalities but are essential for the system's overall performance, security, and usability.

Acceptance Testing

Acceptance Testing ensures that the system meets the expectations of end-users and stakeholders.

- Request Response and Action Confirmation
- Profile and Account Information Updates.

Security Testing

Security Testing ensures that the platform is secure from unauthorized access and that sensitive data is protected.

- Role-Based Access Control (RBAC)
- Data Encryption
- Session Management and Authentication

Performance Testing

- System Load Under Multiple Users
- Search Function Speed and Accuracy

10.3 Unit Testing

Unit Testing focuses on testing individual functions or components to ensure they perform as expected. These tests are typically carried out by developers to validate smaller parts of the code before integrating them into larger systems.

Unit Tests for DeshiTalent:

Test Case ID	Test Case Description	Expected Result
UT01	Test user registration function (valid inputs)	Function should return a success message with user details saved.
UT02	Test user authentication function (valid credentials)	User should be authenticated successfully and return a session token.
UT03	Test password reset function (valid email)	Function should generate a password reset link and send an email.

Table 10.1: Unit Test Details

10.4 Module Testing

Module Testing focuses on testing a specific module or feature. This involves testing the integration of several components that work together to provide a specific service or function.

Module Tests for DeshiTalent:

Test Case ID	Test Case Description	Expected Result
MT01	Test user registration and login module (combining registration & login)	Users should be able to register and log in successfully.
MT02	Test job posting and proposal module (posting jobs and applying)	Jobs should be posted, and freelancers should be able to submit proposals.
MT03	Test payment module (processing payments and escrow management)	Payment should be processed correctly, and funds held in escrow should be released upon project completion.

Table 10.2: Module Test Details

10.4 Acceptance Testing

Acceptance Testing is performed to verify that the system meets the business requirements and user expectations. It ensures the system is ready for deployment and meets all the acceptance criteria set by the stakeholders.

Acceptance Tests for DeshiTalent:

Test Case ID	Test Case Description	Expected Result
AT01	Verify that freelancers can register, create a profile, and bid on jobs.	Freelancers should be able to register, complete their profile, and submit bids on jobs successfully.
AT02	Verify that clients can register, post jobs, and review proposals.	Clients should be able to post jobs, review bids, and accept/reject proposals.
AT03	Verify that payments are processed successfully for completed jobs.	Payment should be processed from the client to the freelancer successfully, with escrow functionality working as intended.
AT04	Verify that the admin can manage users and moderate jobs effectively.	Admin should be able to access all user data, moderate jobs, and generate reports.

Table 10.3: Acceptance Test Details

Unit Test Case: Freelancer Registration

Test Case Name	Unit Test
Test Class	Freelancer Register Controller Test
Test Description	Validation for freelancer name field during registration.
Data Source	User registration form
Test Steps	1. Fill all required fields except the freelancer's name field. 2. Submit the registration form.
Expected Result	An error message should appear indicating that the freelancer's name must not be empty.
Actual Result	A message is displayed: "Freelancer name is required for registration."

Table 10.4: Freelancer Registration Test Cases Planning

Unit Test Case: Job Posting

Test Case Name	Unit Test
Test Class	Job Post Controller Test
Test Description	Validation for job title field during job posting.
Data Source	Job posting form
Test Steps	1. Fill all fields except the job title. 2. Submit the form.
Expected Result	An error message should appear indicating that the job title must not be empty.
Actual Result	A message is displayed: "Job title is required to post a job."

Table 10.5: Client Job Posting Test Cases Planning

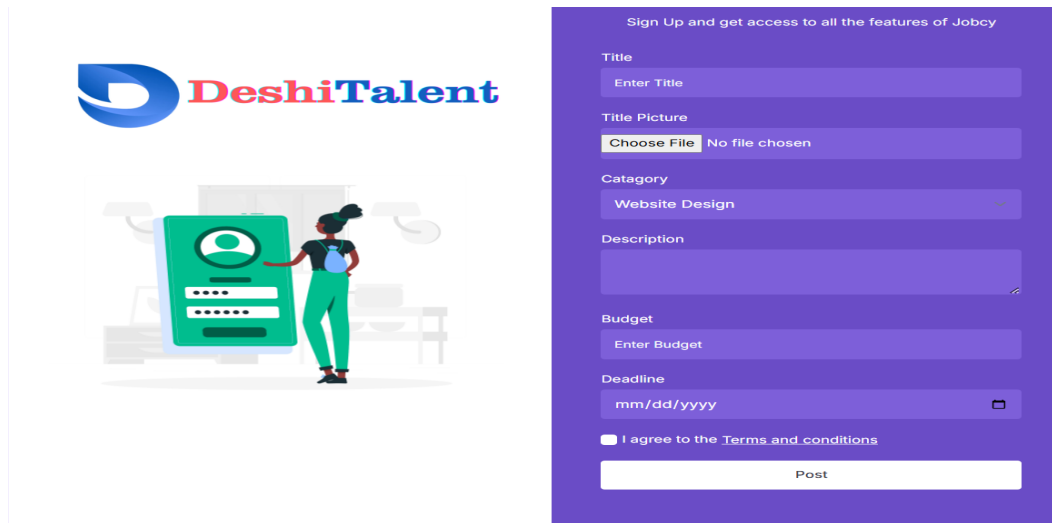


Fig 10.1: After Job Post Test

Unit Test Case: Payment Processing

Test Case Name	Unit Test
Test Class	Payment Calculation Test
Test Description	Ensure correct calculation of payment based on bid amount and admin fee.
Data Source	Payment function parameters
Test Steps	1. Pass a valid bid amount and admin fee percentage to the payment calculation function. 2. Execute the function.
Expected Result	The function should return the correct payment amount.
Actual Result	The function returns the expected payment amount.

Table 10.6: Payment Processing Test Cases Planning

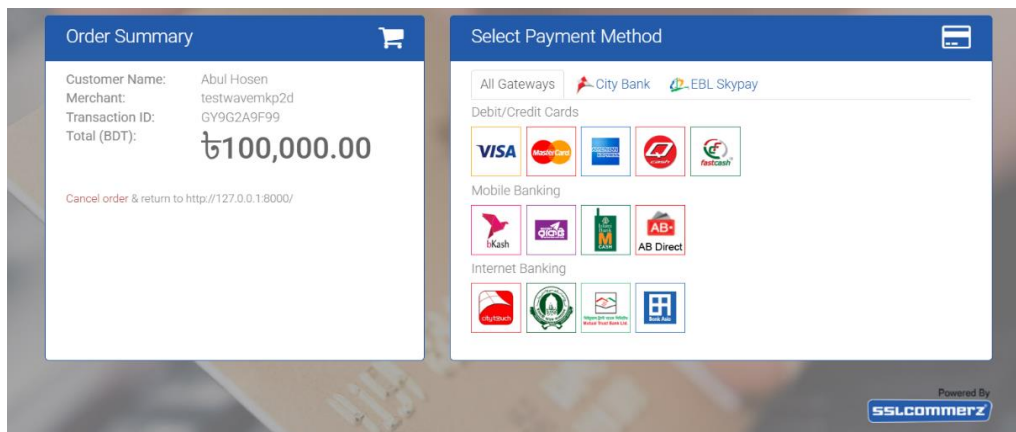


Fig 10.2: Payment Processing Testing

Unit Test Case: Proposal Submission

Test Case Name	Unit Test
Test Class	Proposal Submission Controller Test
Test Description	Validation for proposal description during submission.
Data Source	Proposal submission form
Test Steps	1. Fill all required fields except the proposal description. 2. Submit the proposal form.
Expected Result	An error message should appear indicating that the proposal description must not be empty.
Actual Result	A message is displayed: "Proposal description is required for submission."

Table 10.7: Proposal Submission of Test Cases

Module Test Case: User Registration and Login

Test Case Name	User Registration and Login Module Test
Module Name	User Authentication
Test Description	Validate the complete registration and login process.
Test Steps	1. Register a user with valid data. 2. Try logging in with the registered credentials. 3. Try logging in with invalid credentials.
Expected Result	1. Registration should complete successfully. 2. Login should succeed with valid credentials. 3. Login should fail with invalid credentials and show an error message.
Actual Result	[Result displayed during execution]

Table 10.8: User Registration and Login of Test Cases

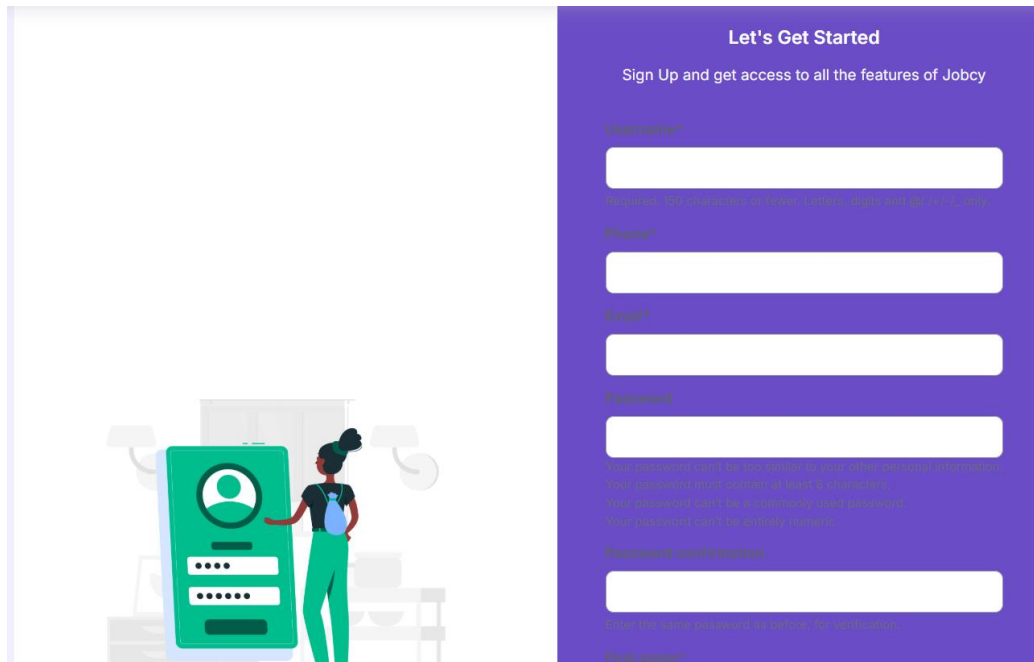


Fig 10.3: Test of Registration

Module Test Case: Job Posting and Proposal Submission

Test Case Name	Job Posting and Proposal Submission Test
Module Name	Job Management
Test Description	Verify that clients can post jobs, and freelancers can submit proposals.
Test Steps	<ol style="list-style-type: none"> 1. Log in as a client and post a new job with valid data. 2. Log in as a freelancer and search for the posted job. 3. Submit a proposal for the job. 4. Ensure the proposal appears on the client’s dashboard.
Expected Result	<ol style="list-style-type: none"> 1. The job should appear in the job listings. 2. The freelancer should be able to submit a proposal. 3. The client should see the submitted proposal on their dashboard.
Actual Result	[Result displayed during execution]

Table 10.9: Module of Proposal Submission Test Cases

Module Test Case: Payment Processing

Test Case Name	Payment Processing Module Test
Module Name	Payment and Escrow Management
Test Description	Validate the payment flow from client to freelancer through escrow.
Test Steps	<ol style="list-style-type: none">1. Log in as a client and hire a freelancer for a job.2. Make a payment to escrow.3. Log in as an admin and verify the payment in escrow.4. Complete the job and release the payment to the freelancer.
Expected Result	<ol style="list-style-type: none">1. Payment should be deducted from the client and appear in escrow.2. The freelancer should receive the payment after job completion.
Actual Result	[Result displayed during execution]

Table 10.10: Module of Payment Processing Test Cases

Module Test Case: Freelancer and Job Search

Test Case Name	Freelancer and Job Search Module Test
Module Name	Search and Filters
Test Description	Validate that users can search for jobs and freelancers using filters.
Test Steps	<ol style="list-style-type: none">1. Log in as a freelancer and search for jobs by category and budget.2. Log in as a client and search for freelancers by skill and rating.3. Verify that the search results match the applied filters.
Expected Result	<ol style="list-style-type: none">1. Search results should be accurate and relevant to the filters applied.2. Both clients and freelancers should be able to find relevant matches.
Actual Result	[Result displayed during execution]

Table 10.11: Module of Freelancer and Job Search Test Cases

10.5 Integration Testing

Integration testing emphasizes verifying those various modules and components of the system function cohesively. The following are the test cases:

Test Case: User Registration and Login Integration

Test Case Name	User Registration and Login Integration Test
Modules Integrated	User Registration Module, Authentication Module
Test Description	Validate the integration of user registration and login functionalities.
Test Steps	<ol style="list-style-type: none">1. Register a user with valid credentials.2. Attempt to log in with the registered credentials.3. Attempt to log in with invalid credentials.4. Verify session management after login.
Expected Result	<ol style="list-style-type: none">1. The user is registered successfully.2. Login is successful with valid credentials.3. Login fails with invalid credentials.4. A session is established for logged-in users.
Actual Result	[Result displayed during execution]

Table 10.12: Integration Testing of User Registration and Login Integration Test Step

Test Case: Job Posting and Proposal Submission Integration

Test Case Name	Job Posting and Proposal Submission Integration Test
Modules Integrated	Job Posting Module, Proposal Submission Module
Test Description	Verify the integration between job posting and proposal submission processes.
Test Steps	<ol style="list-style-type: none">1. Log in as a client and post a job.2. Log in as a freelancer and view the job posting.3. Submit a proposal for the posted job.4. Verify that the client sees the proposal on their dashboard.
Expected Result	<ol style="list-style-type: none">1. Jobs are visible to freelancers after posting.2. Freelancers can submit proposals.3. Clients can view submitted proposals.
Actual Result	[Result displayed during execution]

Table 10.13: Integration Testing of Job Posting and Proposal Submission Integration

Chapter 11 - Deployment

10.1 Core module code snippet

Login

```
def user_login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            user = authenticate(
                request, username=cd["username"], password=cd["password"])
            if user is not None:
                if user.is_freelancer:
                    login(request, user)
                    list = Freelancer.objects.filter(user=request.user)
                    print(list)
                    return redirect('dashboard')
                elif user.is_client:
                    login(request, user)
                    list = Client.objects.filter(user=request.user)
                    print(list)
                    return redirect('clientdashboard')
                else:
                    return HttpResponse("Diable")
            else:
                return HttpResponse("Invalid Login")
        else:
            form = LoginForm()
    return render(request, 'sign-in.html', {'data': form})
```

Fig 11.1: Login Code of Django (Python-view.py)

Client Profile

```
def ClientProfileView(request):
    if request.user.is_client:
        profile = Client.objects.get(user_id=request.user.id)

        if request.method == 'POST':
            about = request.POST.get('about')
            company = request.POST.get('company')
            country = 'Bangladesh'
            language = request.POST.get('language')
            image = request.FILES.get('image')

            client_profile = ClientProfile.objects.filter(client=profile).first()

            if client_profile:
                client_profile.about = about
                client_profile.company = company
                client_profile.country = country
                client_profile.language = language
                client_profile.image = image
                client_profile.save()
                messages.success(request, 'Profile updated successfully!')
                return HttpResponseRedirect('Profile updated successfully!')
            else:
                client_profile = ClientProfile.objects.create(
                    client=profile,
                    about=about,
                    company=company,
                    country=country,
                    language=language,
                    image=image
                )
                messages.success(request, 'Profile created successfully!')
                return HttpResponseRedirect('Profile created successfully!')
        else:
            context = {'freelancer_profile': profile}
            return render(request, 'test-client-profile.html', context)
    else:
        return render(request, 'test-client-profile.html')

# Client Account (Registration)
class ClientRegistrationView(CreateView):
    model = User
    form_class = ClientSignupForm
    template_name = './sign-up.html'

    def get_context_data(self, **kwargs):
        kwargs['user-type'] = 'client'
        return super().get_context_data(**kwargs)

    def form_valid(self, form):
        email = form.cleaned_data['email']
        name = form.cleaned_data['username']

        user = form.save()
        verification_code = sent_email_verification(email, name)
        verification_instance = VerificationModel(
            code=verification_code, user_id=user.id)
        verification_instance.save()

        login(self.request, user)
        return redirect('verification')
```

Fig 11.2: Client Profile Code of Django (Python-view.py)

Freelancer

```
class FreelancerView(CreateView):
    model = User
    form_class = FreelancerSignUpForm
    template_name = './sign-up.html'

    def get_context_data(self, **kwargs):
        kwargs['user-type'] = 'freelancer'
        return super().get_context_data(**kwargs)

    def form_valid(self, form):
        email = form.cleaned_data['email']
        name = form.cleaned_data['first_name']
        print(f'Registration Name : {name} Email :{email}')
        user = form.save()
        verification_code = sent_email_verification(email, name)
        verification_instance = VerificationModel(
            code=verification_code, user_id=user.id)
        verification_instance.save()

        login(self.request, user)
        return redirect('verification')

@login_required
def FreelancerProfileView(request):
    if request.user.is_freelancer:
        profile = Freelancer.objects.get(user_id=request.user.id)
        freelancer_profile = FreelancerProfile.objects.get(freelancer=profile)
        if request.method == 'POST':
            about = request.POST.get('about')
            skill = request.POST.get('skills')
            interest = request.POST.get('interest')
            image = request.FILES.get('image')

            freelancer_profile = FreelancerProfile(
                freelancer=profile,
                about=about,
                skill=skill,
                interest=interest,
                image=image
            )
            freelancer_profile.save()
            messages.success(request, 'Freelancer profile updated successfully!')
            return HttpResponseRedirect(reverse('FreelancerProfileEdit'))
        else:
            context = {'freelancer_profile': freelancer_profile}
            return render(request, 'test-profile.html', context)
    else:
        return render(request, 'test-profile.html')
```

Fig 11.3: Freelancer of Django (Python-view.py)

Dashboard

```
@login_required
def dashboard(request):
    user_details = None
    project_details = None

    user_data = user_profile(request)

    if request.user.is_freelancer:
        user_details = Freelancer.objects.get(user=request.user)
        print(user_details.user_id)
        freelancer_application = ApplicationModel.objects.filter(
            freelancer_id=user_details.user_id)
        print(freelancer_application)

        project_details = []
        for application in freelancer_application:
            project = ProjectModel.objects.get(id=application.project_id)
            project_details.append(project)

    context = {'details': user_details,
              'user_data': user_data,
              'project_details': project_details}

    return render(request, 'dashboard.html', context)
```

Fig 11.4: Dashboard Code of Django (Python-view.py)

Homepage of Python code

```
def home(request):
    projectdata = ProjectModel.objects.all()[:5]

    if not request.user.is_authenticated:
        print('Not User')
        return render(request, 'home.html', {'data': projectdata}) # Handle anonymous user

    elif request.user.is_freelancer:
        print('Freelancer')
        user_data, projectdata = homepage_freelancer(request)

    elif request.user.is_client:
        user_data, projectdata = homepage_client(request)
        print('Client')
        print(projectdata)

    return render(request, 'home.html', {'data': projectdata, 'user_data': user_data})
```

Fig 11.5: Homepage Code of Django (Python-view.py)

Authentication code

```
@login_required
def verification_account(request):
    if request.method == 'POST':
        forms = VerificationForm(request.POST)
        # print(forms)
        if forms.is_valid():
            code = forms.cleaned_data['code']
            code_instance = VerificationModel.objects.get(
                user_id=request.user.id)

            if code_instance.is_verify == False:
                if code_instance.code == int(code):
                    code_instance.is_verify = True
                    code_instance.save()
                    return HttpResponseRedirect('Verification Successfull!')
                elif code_instance.code != int(code):
                    return HttpResponseRedirect("Code Error!")
            else:
                return HttpResponseRedirect("Already Verified!")
    return render(request, 'sign-up-verify.html')
```

Fig 11.6: Authentication Code of Django (Python-view.py)

Posting Project

```
@login_required
def NewProjectPostView(request):
    if request.method == 'POST':
        form = ProjectForm(request.POST, request.FILES)
        print('Check line 225')
        if form.is_valid():
            # Extract form data
            print('Check line 227')
            title = form.cleaned_data['title']
            image = form.cleaned_data['image']
            catagories = form.cleaned_data['catagories']
            description = form.cleaned_data['description']
            budget = form.cleaned_data['budget']
            deadline = form.cleaned_data['deadline']

            client_instance = Client.objects.get(user=request.user)

            project = ProjectModel(
                client_id=client_instance,
                title=title,
                image=image,
                catagories=catagories,
                description=description,
                budget=budget,
                deadline=deadline
            )
            print('Check line 248')
            project.save()

            return HttpResponseRedirect('Successfully Posted')
        else:
            print('Check line 253')
            form = ProjectForm()
            return render(request, 'job-post.html', {'projectform': form})
```

Fig 11.7: Posting Project Code of Django (Python-view.py)

Client Manage Project

```
@login_required
def ApproveProjectFreelancer(request, freelancer_id, project_id):
    if request.user.is_client:
        project = ProjectModel.objects.get(id=project_id)
        freelancer = Freelancer.objects.get(user_id=freelancer_id)
        if ApproveProject.objects.filter(
            freelancer_id=freelancer.user_id,
            project_id=project.id
        ).exists():
            # Display an error message to the user
            return HttpResponse('The freelancer has already been approved/Reject for this project.')
        else:
            approve_project = ApproveProject(
                client_id=request.user.id,
                project_id=project.id,
                freelancer_id=freelancer.user_id,
                status= True )
            approve_project.save()
            email = freelancer.email
            name = freelancer.first_name + ' ' + freelancer.last_name
            project_title = project.title
            sent_notification(email, name, project_title)
            return HttpResponse('Successfully!')
    return render(request, 'test-candidate-list.html')

#Client Reject the Application

@login_required
def RejectProjectFreelancer(request, freelancer_id, project_id):
    if request.user.is_client:
        project = ProjectModel.objects.get(id=project_id)
        freelancer = Freelancer.objects.get(user_id=freelancer_id)
        if ApproveProject.objects.filter(
            freelancer_id=freelancer.user_id,
            project_id=project.id
        ).exists():
            # Display an error message to the user
            return HttpResponse('The freelancer has already been approved/Reject for this project.')
        else:
            approve_project = ApproveProject(
                client_id=request.user.id,
                project_id=project.id,
                freelancer_id=freelancer.user_id,
                status= False)
            approve_project.save()
            email = freelancer.email
            name = freelancer.first_name + ' ' + freelancer.last_name
            project_title = project.title
            sent_notification(email, name, project_title)
            return HttpResponse('Successfully!')
    return render(request, 'test-candidate-list.html')

@login_required
def AcceptFreelancerList(request, project_id):
    if request.user.is_client:
        accept_details = ApproveProject.objects.get(project_id=project_id)
        user_details = Freelancer.objects.get(user_id=accept_details.freelancer_id)
        project_details = ProjectModel.objects.get(id=project_id)

    return render(request, 'test-accept-list.html',
        {'data':user_details, 'projectdata':project_details})
```

Fig 11.8: Client Management Project Code of Django (Python-view.py)

SSLCommerz Payment Backend code

```
import string
import random
from django.conf import settings
from django.contrib.auth.decorators import login_required
from django.http import HttpRequest,HttpResponse
from sslcommerz_lib import SSLCOMMERZ

def unique_id(size=10,chars=string.ascii_uppercase+string.digits):
    return ''.join(random.choice(chars) for _ in range(size))

def SSL_Payment(user_id,name,number,email,amount,address,id,item):
    settings = { 'store_id': 'wavem655a4206f322e',
                 'store_pass': 'wavem655a4206f322e@ssl',
                 'issandbox': True }
    sslcommez = SSLCOMMERZ(settings)

    post_body = {}
    post_body['total_amount'] = amount
    post_body['currency'] = "BDT"
    post_body['tran_id'] = id

    post_body['success_url'] = "http://127.0.0.1:8000/payment/success"
    post_body['fail_url'] = "http://127.0.0.1:8000/"
    post_body['cancel_url'] = "http://127.0.0.1:8000/"
    post_body['emi_option'] = 0

    post_body['cus_name'] = name
    post_body['cus_email'] = email
    post_body['cus_phone'] = number
    post_body['cus_add1'] = address
    post_body['cus_city'] = "Dhaka"
    post_body['cus_country'] = "Bangladesh"

    post_body['shipping_method'] = "NO"
    post_body['multi_card_name'] = ""
    post_body['num_of_item'] = 1
    post_body['product_name'] = item
    post_body['product_category'] = "Digital Product"
    post_body['product_profile'] = "general"

    # Additional Information
    post_body['value_a'] = amount
    post_body['value_b'] = item
    post_body['value_c'] = user_id
    post_body['value_d'] = id

    response = sslcommez.createSession(post_body)

    # print(response)
    return 'https://sandbox.sslcommerz.com/gwprocess/v4/gw.php?Q=pay&SESSIONKEY=' +
response["sessionkey"]
```

Fig 11.8: SSLCommerz Payment Code of Django (Python-view.py)

Chapter 12 - Implementation

12.1 Process of Implementation

Thorough training is essential to ensure that users are well acquainted with the features and functionality of the developed system. Once the system is fully functional, it is crucial to ensure that users or operators have the necessary skills and knowledge to use the system efficiently. The following chart outlines the specific training procedures designed to achieve this objective for *DeshiTalent – Bangladeshi Talent Hub*. The table details the users, their respective training needs, estimated timeframes, and any pertinent remarks related to the training process.

SL	User	Training Scope	Time Period	Comment
1	Freelancer	<ul style="list-style-type: none"> - Platform navigation: Account setup, profile creation, bidding on projects. - Payment processes (e.g., escrow system, withdrawals). - Skill assessment and certifications. 	1-2 days	Online video tutorials and help center articles can be provided.
3	Client	<ul style="list-style-type: none"> - Project posting and defining job requirements. - Reviewing proposals and hiring freelancers. - Setting milestones and managing payments. 	1-2 days	Provide a step-by-step guide for posting and managing projects.
5	Admin	<ul style="list-style-type: none"> - Managing user accounts (approvals, rejections, verification). - Monitoring disputes and resolving conflicts. - Using admin dashboard for reports and analytics. 	2-3 days	Onboarding sessions with detailed instructions on platform tools.
6	Admin	<ul style="list-style-type: none"> - System security and data protection protocols. - Handling escalations and ensuring compliance with platform policies. 	1-2 days	Include periodic refresher sessions for updates.

Table 12.1: Process of Implementation

Chapter 13 - Critical Appraisal and Evaluation

13.1 Objective

The primary objective of **DeshiTalent** is to create a robust and user-friendly **freelancer marketplace** that connects **freelancers** with **clients** in Bangladesh. The platform aims to empower local talent and provide businesses with access to skilled professionals, offering an efficient and secure way to collaborate on projects.

Specific Objectives:

- ✓ **Facilitate Freelancer-Client Connections**
- ✓ Promote Local Talent and Opportunities
- ✓ Ensure Safe and Secure Transactions with local payment system
- ✓ Easy to access the Project Bidding system

Objective 1

DeshiTalent aims to function as a dynamic online platform that facilitates seamless connections between skilled freelancers and discerning clients. By enabling clients to effortlessly discover and engage with qualified professionals across a diverse spectrum of industries, including information technology, design, writing, and marketing, the platform aims to streamline the hiring process. Moreover, DeshiTalent empowers freelancers to effectively showcase their unique skillsets, comprehensive portfolios, and in-depth expertise, thereby enhancing their visibility and increasing their prospects of securing lucrative employment opportunities.

Objective 2

DeshiTalent envisions fostering a thriving ecosystem that nurtures and elevates Bangladeshi talent on both domestic and international platforms. By providing a comprehensive suite of services, including specialized training programs, industry-recognized certifications, and rigorous skill assessments, DeshiTalent aims to empower freelancers to enhance their professional development and achieve career growth. Furthermore, the platform actively encourages businesses to prioritize local talent acquisition, thereby stimulating economic growth within Bangladesh and creating a sustainable pathway for talent development within the country.

Objective 3

DeshiTalent prioritizes secure and transparent financial transactions by implementing a robust escrow-based payment system that safeguards the interests of both freelancers and clients. To facilitate seamless payments, the platform integrates a diverse range of local and international payment gateways, including bKash and Local Bank with SSLCommerz offering users flexibility and convenience. Furthermore, DeshiTalent establishes a clear and equitable dispute resolution process, ensuring that any potential conflicts are addressed promptly and fairly.

13.2 Objective Not Fully Achieved

While most objectives for the **DeshiTalent** platform have been successfully addressed, one key feature, **the mobile application for tracking location**, was not fully implemented. This was the only objective that could not be completed within the defined project scope.

Reasons for non-achievement

The primary reasons this feature was not implemented are:

1. **Limited Time Frame:** The project had a very brief timeline to complete the tasks, making it challenging to develop and integrate a fully functional mobile app within the given period.
2. **Lack of Continuous Hosting Server:** A mobile application for location tracking requires a live, continuous hosting server. The absence of this infrastructure made it difficult to implement this feature, as the app would not function properly without a reliable server connection.

What Could Have Been Done

To achieve the location tracking feature, the following adjustments could have been made:

1. **Rescheduling the Timeline:** Extending the project timeline would have allowed for additional time to complete the mobile application development, including the location tracking feature.
2. **Learning Hybrid Application Development:** In order to build a mobile app that works across multiple platforms (iOS and Android), investing time in learning **hybrid application development** would have been essential. This approach would have enabled the creation of the mobile app within the time constraints, using frameworks **Flutter**.

Chapter 14 - Lessons Learned

14.1 Pre-project Review Closing

DeshiTalent is a proposed web-based platform designed to serve as a dedicated freelance marketplace for the Bangladeshi market. The project will adhere to a comprehensive academic framework, including proposal submission, title defense, development, and detailed documentation throughout the process.

DeshiTalent aims to connect Bangladeshi freelancers with clients, fostering the development of local talent. The platform will provide businesses with easy access to skilled professionals and offer freelancers a diverse range of employment opportunities within the Bangladeshi market.

14.2 What I have Learnt

The *DeshiTalent* project afforded me valuable technical and business insights. On the technical front, I developed proficiency in backend development utilizing Django, successfully integrated secure payment gateways such as bKash and SSLCommerz and gained practical experience with SQLITE3 for database management. Furthermore, I honed my front-end development skills by employing React.js and Tailwind CSS to enhance the user interfaces for freelancers, clients, and administrators, with a keen interest in further investigating hybrid frameworks like React Native and Flutter. This hands-on experience provided me with a comprehensive understanding of the technical intricacies involved in creating a secure, scalable, and efficient platform.

From a business standpoint, I gained valuable knowledge regarding the dynamics of the freelance market, emphasizing the critical importance of fostering trust between freelancers and clients. I also acquired insights into establishing sustainable revenue streams for the platform. Moreover, I gained valuable knowledge in user engagement, customer support, and brand development. This project underscored the significance of thorough market research and the necessity of developing a scalable business model to effectively address evolving market demands.

14.3 The problems I have faced

The development phase of the DeshiTalent project presented several significant challenges. Designing a robust and scalable database schema to effectively manage complex relationships between various entities, including freelancers, clients, projects, and payments, proved demanding. Implementing all planned features, the project's tight timeline and limited resources for continuous server hosting. Furthermore, developing a responsive and intuitive user interface that seamlessly integrated diverse functionalities for different user roles while maintaining consistency across multiple devices presented a significant challenge, necessitating iterative refinement and careful attention to detail. These challenges necessitated meticulous problem-solving and prioritization throughout the development process.

14.4 What Solution Occurred

To address challenges, database design was optimized, core features were prioritized, and a mobile-first design approach was adopted. I prioritized core features like **secure payments**, **project management**, and **user profiles** and Dashboard. User testing facilitated iterative refinements, ensuring a user-friendly experience.

Chapter 15 - Conclusion

15.1 Summary of the Project

DeshiTalent – Bangladeshi Freelancer Hub is an innovative online platform designed to connect freelancers with clients, fostering a dynamic freelancing ecosystem. The platform addresses key requirements for freelancers, clients, and administrators, enabling project posting, bidding, secure payments, real-time communication, and efficient management tools. Integration with SSLCommerz ensures a secure and seamless payment process, while features like rating and review systems enhance trust and transparency. Developed using Python, Django, SQLite3, Tailwind CSS, REST APIs, and other modern technologies, DeshiTalent is designed to be scalable, user-friendly, and robust. This documentation provides comprehensive insights into the system's design, implementation, and features.

15.2 Project Goal

The primary objective of this project was to create a centralized platform that bridges the gap between freelancers and clients, ensuring secure, transparent, and efficient transactions. The goals of the project include:

- ✓ Establish a reliable system for project posting and bidding.
- ✓ Providing secure payment solutions with an escrow mechanism to protect both parties.
- ✓ Enabling efficient communication and collaboration through dashboards and messaging systems.
- ✓ Incorporating rating and review features to build trust and transparency among users.
- ✓ Empowering administrators with tools for platform management, analytics, and dispute resolution.

15.3 Success of the Project

The project successfully achieved its core objectives, fulfilling the criteria established at its inception. Freelancers can showcase their expertise, clients can find the talent they need, and secure payments enhance user confidence. While the development of a mobile application for location tracking was deferred due to time and resource constraints, the project is otherwise complete and successful in its goals.

15.4 Documentation Contribution

This documentation thoroughly covers all aspects of the project, from its initiation to its completion. It includes detailed illustrations, timeboxing techniques, system architecture, and a thorough analysis of goals and deliverables. Every necessary detail for understanding and implementing the project has been systematically provided.

15.5 Value of the Project

DeshiTalent offers significant value to the freelancing industry in Bangladesh by providing a structured, secure, and user-friendly platform. It simplifies the freelancing process for both clients and freelancers while ensuring accountability and transparency. This project serves as a benchmark for enhancing local talent accessibility and fostering economic growth in the freelancing sector. The skills and insights gained during the development process provide a solid foundation for future professional endeavors and system enhancements.

15.6 Personal Experience

The development of DeshiTalent has been an invaluable learning experience. I encountered and resolved numerous technical and managerial challenges, enhancing my expertise in problem-solving, time management, and systems integration. I gained proficiency in integrating secure payment gateways, designing REST APIs, and developing user-centric features. This experience has greatly enriched my technical skills and prepared me for future professional challenges in software development.

References

<https://www.researchgate.net/publication/353804019> The Gig Economy A Case Study Analysis of Freelancer.com

<https://www.thedailystar.net/opinion/views/news/bangladeshs-reputation-freelancing-powerhouse-the-line-3620261>

Oxford Internet Institute (OII). (2017)

Forbes.

<https://www.forbes.com/sites/rachelwells/2024/08/04/top-10-countries-hiring-freelancers-in-2024-from-research/>

Freelancer.com.

<https://www.geeksforgeeks.org/>

<https://www.tutorialspoint.com/index.htm>

Plagiarism Report

211-16-555

ORIGINALITY REPORT

6%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1

dspace.daffodilvarsity.edu.bd:8080

Internet Source

3%

2

Submitted to Daffodil International University

Student Paper

1%

3

Submitted to University of Wales Institute,
Cardiff

Student Paper

<1%

4

Submitted to South Thames College

Student Paper

<1%

5

www.linode.com

Internet Source

<1%

6

Mara L. Daiker. "The CPHIMS Review Guide -
Preparing for Success in Healthcare
Information and Management Systems",
Routledge, 2021

Publication

<1%

7

Submitted to Southern New Hampshire
University - Continuing Education

Student Paper

<1%