SKILLBRIDGE:A STUDENT TALENT MARKETPLACE PLATFORM

A Mini Project Report

submitted by

ANSILA NV

(MES24MCA-2007)

to the APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering

Kuttippuram, Malappuram – 679582

October, 2025

Declaration

I undersigned hereby declare that the project report **SKILLBRIDGE:A STUDENT TALENT MARKETPLACE PLATFORM** submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Mr. Balachandran K P, Associate Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

ANSILA NV

(MES24MCA-2007)

08/10/2025

DEPARTMENT OF COMPUTER APPLICATIONS

MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **SKILLBRIDGE:A STUDENT TALENT MARKETPLACE PLATFORM** is a bonafide record of the Mini Project work during the year 2025-26 carried out by **ANSILA NV (MES24MCA-2007)** submitted to the APJ Abdul Kalam Technological University, in partial fulfilment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor

Head of The Department

Acknowledgment

I would like to begin by expressing my deep sense of gratitude to the Almighty, whose blessings and guidance have been a constant source of strength and inspiration throughout the completion of my project work titled “SKILLBRIDGE: A STUDENT TALENT MARKETPLACE PLATFORM”, undertaken as part of the requirements for the degree of Master of Computer Applications at MES College of Engineering, Kuttippuram, under APJ Abdul Kalam Technological University, Kerala.

I owe my sincere thanks to my project guide, Mr. Balachandran K P, Associate Professor, Department of Computer Applications, MES College of Engineering, Kuttippuram, for her valuable guidance, constructive suggestions, and constant encouragement at every stage of the project. Her insights and expertise have been instrumental in shaping this work into its final form.

I also express my heartfelt gratitude to Professor Hyderali K Head of the Department of Computer Applications, for providing me with the necessary facilities, encouragement, and academic support during the course of my project. My sincere thanks also go to all the faculty members and technical staff of the Department of Computer Applications for their timely advice, cooperation, and moral support.

I am especially thankful to my friends and classmates for their valuable feedback, collaboration, and encouragement, which motivated me to enhance the quality of this work.

Finally, I express my deepest gratitude to my family members for their unconditional love, patience, and constant support, which gave me the confidence to successfully complete this project

ANSILA NV

(MES24MCA-2007)

Abstract

SKILLBRIDGE: A STUDENT TALENT MARKETPLACE PLATFORM is a modern web-based platform developed to connect skilled students with clients both within and beyond their academic institutions. It provides a structured, secure, and easy-to-use environment where students can showcase and monetize their creative and technical talents—such as crafting, baking, photography, and drawing—while gaining real-world experience. Traditional methods of finding work, including social media, personal contacts, or word-of-mouth, are often unorganized, unreliable, and lack proper verification and payment security. This results in limited opportunities for students and makes it difficult for clients to find trustworthy talent.

The platform addresses these challenges by offering a dedicated digital ecosystem with three main modules: Admin, Service Provider (Student), and User (Client). The Admin module manages the platform by verifying services, monitoring users, handling complaints, and maintaining system integrity. The Service Provider module allows students to register, create and manage portfolios, handle client requests, set pricing, communicate with clients, and receive feedback and ratings. The User module enables clients to browse services, submit requests, communicate with students, track progress, make secure payments, and provide feedback or complaints.

Developed using modern web technologies—including HTML, CSS, Bootstrap, JavaScript for the front end, Django for the back end, and MySQL for database management—SkillBridge ensures secure data handling, smooth performance, and scalability. The platform integrates real-time communication, centralized request management, and reliable payment processing, creating a safe and organized freelancing ecosystem.

In conclusion, SkillBridge provides a professional and reliable marketplace for students to showcase their talents, earn income, and gain experience, while enabling clients to access verified, high-quality services in a secure and organized environment.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

In the present digital era, students possess a wide range of creative and technical skills that can be transformed into valuable services. However, many talented students struggle to find suitable platforms to showcase their abilities or connect with clients who require their expertise. Most rely on unorganized channels such as social media or personal contacts, which lack reliability, proper verification, and secure payment systems. This often limits their opportunities for professional growth and real-world experience.

SkillBridge is a web-based platform developed to solve this issue by providing a dedicated and secure space where students can display their skills and clients can easily find and hire them. The system simplifies the process of service discovery, booking, and communication, ensuring that both students and clients can interact efficiently.

The platform includes three main modules—Admin, Service Provider (Student), and User (Client). Each module is designed with specific features to manage registration, service listings, requests, payments, and feedback. The Admin oversees platform operations and ensures security and reliability.

SkillBridge is developed using HTML, CSS, Bootstrap, and JavaScript for the frontend, and Python (Django) with MySQL for the backend. The platform emphasizes user-friendly design, security, and smooth data management.

By providing an organized online environment, SkillBridge encourages students to gain professional exposure, earn income, and enhance their skills, while helping clients access quality services easily and confidently.

1.1 Motivation

Many students have creative and technical skills like designing, programming, and photography, but they often lack a proper platform to showcase their talents or find clients. Most depend on social media or personal contacts, which are unorganized and unreliable. This makes it difficult for students to gain experience or earn income from their skills.

The SKILLBRIDGE: A STUDENT TALENT MARKETPLACE PLATFORM system is developed to solve this problem by creating an online platform where students can display their work and connect with people who need their services. It helps students improve their confidence, gain real-world experience, and earn money safely.

For clients, the platform offers an easy way to find skilled students for different tasks at affordable prices. It also ensures security, transparency, and smooth communication between both sides.

Technically, the project uses HTML, CSS, Bootstrap, JavaScript, Python (Django), and MySQL to provide a user-friendly and reliable system that bridges the gap between student talent and client needs.

1.2 Objectives

The main problem this mini project addresses is that students do not have a proper platform to show their skills and connect with clients. Right now, students rely on social media, friends, or word-of-mouth to find work. These methods are unreliable, unorganized, and sometimes unsafe. Because of this, students have low visibility, find it hard to get real clients, struggle to manage requests, and have no proper system for payments or feedback. As a result, students miss opportunities to earn money and gain experience, and clients have difficulty finding trustworthy talent.

This project assumes that SkillBridge, a safe and organized platform for student talents, will help students show their skills in a professional way and connect with clients securely. Clients will be able to find verified and talented students in one place, making it easier to hire them. With features like direct chat, secure payments, and feedback, SkillBridge will make transactions

smooth, maintain good service quality, and create a reliable freelancing environment for students and clients.

The objectives of the project can be summarized as follows:

1. To build a web and mobile platform that helps students connect with clients easily.
2. To allow students to sign up as service providers, show their work, manage requests, set prices, and communicate with clients.
3. To let clients browse services, make requests, track progress, pay securely, and give feedback or ratings.
4. To provide an admin panel to check services, monitor users, handle complaints, and keep the platform running smoothly.
5. To include real-time chat so students and clients can communicate easily.
6. To create a safe, organized, and campus-focused freelancing system that increases student visibility, helps them earn money, and keeps clients satisfied.

1.3 Contributions

- Centralized Talent Platform – A single place for students to show their skills and find clients.
- Real-Time Service Requests – Students and clients can see updates immediately.
- Secure Transactions – Payments are safe and reliable.
- Role-Based Dashboards – Separate panels for students, clients, and admins.
- Feedback and Ratings – Clients can give feedback to maintain quality.
- Conflict-Free Scheduling – Manages multiple requests without overlap.
- Scalable Architecture – Built with Django and MySQL to support growth.
- Enhanced User Experience – Makes the platform easy and convenient for students and clients.

1.4 Report Organization

The project report is organized into several chapters to provide a clear and systematic presentation of the work carried out. **Chapter 2** focuses on the system study, describing the existing system and its limitations, followed by a detailed explanation of the proposed system and its functionalities. **Chapter 3** covers the methodology used for implementation, outlining the design approach, software tools, and a detailed description of the various modules, along with sprint details**. Chapter 4** presents the results and discussions, showcasing the implementation outcomes, screenshots of the developed system, and an analysis of the results in relation to the project objectives. Finally, **Chapter 5** summarizes the conclusion and future work, highlighting the overall contributions, benefits of the developed system, and suggesting possible directions for future enhancements.

Chapter 2. System Study

SkillBridge is a student talent marketplace platform that connects students with clients in and outside their schools. It lets students showcase and earn from skills like crafting, baking, photography, and drawing while gaining experience. The platform has three parts: Admin, which manages the system and handles complaints; Service Provider (Student), which lets students register, share work, manage requests, set prices, chat, and see feedback; and User (Client), which lets clients browse services, make requests, chat, pay, and give feedback. SkillBridge provides a safe and organized space for students to earn and for clients to find reliable talent.

2.1 Existing System

Presently, students with creative or technical skills often rely on informal ways to find clients, like social media, personal contacts, or word-of-mouth. These methods are unreliable, do not verify skills properly, and lack secure payment options. Clients also find it hard to identify genuine and skilled students. There is no central system to manage services, communication, or feedback between students and clients. As a result, many students cannot show their talents or earn from their skills, and clients miss out on finding the right talent.

2.2 Proposed System

The proposed system, SkillBridge, is a safe and organized online platform for students to show their talents and connect with clients. Students can register, list their skills, manage requests, and chat directly with clients. Clients can easily browse services, make bookings, track progress, and give feedback or ratings after the work is done.

2.3 Functionalities of Proposed System

- **Secure Registration & Login** – Students and clients can create accounts and access the platform safely.
- **Service Listing** – Students can list their skills and services for clients to view.
- **Service Search & Booking** – Clients can search for services and book requests easily**.**
- **Real-Time Request Updates** – Students and clients can track request progress instantly
- **Secure Payments** – Clients can make payments safely through the platform.
- **Portfolio & Pricing Management** – Students can manage their profiles, services, and pricing.
- **Feedback & Ratings** – Clients can give feedback and ratings to maintain service quality**.**
- **Integrated Chat** – Students and clients can communicate directly through the platform.
- **Admin Dashboard** – Admins manage users, verify services, and handle complaints.
- **Scalable Platform** – Built with modern technologies to support growth and efficient operation.

Chapter 3. Methodology

Developing a reliable and scalable web-based student talent marketplace requires a structured approach to ensure timely delivery, adaptability to evolving requirements, and high-quality software. For the SkillBridge : A Student Talent Marketplace Platform, the Agile methodology was adopted to guide the development process. Agile encourages iterative development, continuous feedback, and close collaboration between developers and stakeholders, making it ideal for dynamic systems like SkillBridge where user needs and features may evolve during the project.

3.1 Introduction

The Agile methodology is an iterative and incremental approach to software development that emphasizes adaptability, collaboration, and continuous improvement. Unlike traditional models, Agile promotes regular feedback from stakeholders and flexible responses to changing requirements. Development is organized into sprints, with each sprint delivering a functional part of the system.

This methodology is well suited for SkillBridge: A Student Talent Marketplace Platform, where features like service listing, booking requests, real-time chat, and payment processing may evolve based on user feedback. Agile enables developers to refine modules such as student registration, client dashboards, and admin management efficiently. It also supports frequent testing and review, ensuring high software quality and timely delivery of user-friendly features that meet the dynamic needs of students and clients in a digital talent marketplace.

3.2 Software Tools

The following software tools and technologies were used to develop the project Skillbridge:A Student Talent Marketplace Platform,

Table 3.2: List the software tools or languages used for the project development

Operating System	Windows 10
Front End	HTML, CSS, JavaScript,Bootstrap
Back End	Python (Django)
Framework	Django
Database	MySQL
IDE	Android Studio/Jet Brains Pycharm
Version Control	Git & GitHub

3.2.1 Django

Django was chosen as the backend framework for SkillBridge: A Student Talent Marketplace Platform due to its robust, secure, and scalable architecture. It follows the Model-View-Template (MVT) pattern and includes built-in features for authentication, URL routing, and database management. Since SkillBridge: A Student Talent Marketplace Platform involves multiple users—students, clients, and admins—accessing and managing services, requests, and payments simultaneously, Django's ORM and middleware support ensure smooth performance and easy maintainability. Its Python-based ecosystem also allows rapid development and integration with third-party libraries for features like chat, notifications, and reporting.

3.2.2 MySQL

MySQL was chosen as the database for its reliability, structured data handling, and support for relational integrity. It stores important information such as student profiles, service details, booking requests, payments, and login credentials. MySQL's ability to handle complex queries and transactions ensures data consistency and security, which are vital for SkillBridge: A Student

Talent Marketplace Platform. Its compatibility with Django's ORM further streamlines backend development and data operations.

3.3 Module Description

SkillBridge: A Student Talent Marketplace Platform is divided into three main modules: Service Provider (Student) Module, User (Client) Module, and Admin Module. Each module is designed to handle specific responsibilities and streamline the process of managing services, requests, and interactions for its respective user type.

3.3.1 Service Module

This module allows students to interact with the system and manage their services efficiently.

- Student Registration/Login – This feature enables students to securely create accounts and log in to the platform, providing safe access to their personalized dashboard and service management tools.
- Manage Services/Works – Students can add new services they offer, update details of existing services, or remove services they no longer provide, helping keep their portfolio accurate and appealing.
- View Client Requests – Students can view all incoming requests from clients, allowing them to stay informed about demand and respond promptly to potential job opportunities.
- Update Request Status – This functionality lets students accept client requests, update the progress as they work on them, and mark services as completed, ensuring clear communication and tracking.
- Chat with Clients – Students can communicate directly with clients through an integrated chat system, enabling quick clarification of requirements and building strong client relationships.
- Set or Update Pricing – Students have the ability to define or modify the cost of their services at any time, allowing flexibility to adjust prices based on effort, complexity, or market demand
- View Feedback & Ratings – After service completion, students can access client reviews and ratings, which help them assess performance and improve the quality of their offerings.
- Send Complaints & View Admin Replies – Students can report any issues or concerns they encounter, and the admin responds through the platform, ensuring that problems are addressed efficiently and fairly.

3.3.2 User Module

This module enables clients to manage their service requests and interact with students

- Registration – This feature allows clients to create an account to access all platform features securely and manage their service requests.
- Login – Clients can securely log in to their accounts to access personalized dashboards and service management tools.
- View Product & Request – Clients can browse available student services and submit requests based on their specific needs.
- Chat – Clients can communicate directly with students through an integrated chat system to discuss service details and clarify requirements.
- View Request Status & Make Payment – Clients can track the progress of their service requests in real-time and make secure payments upon completion.
- Send Feedback & Rating – After service delivery, clients can provide feedback and ratings to help improve service quality and student reputation.
- Send Complaint & View Reply – Clients can report any issues or concerns through the platform and receive timely responses from the admin to resolve them effectively

3.3.3 Admin Module

This module provides administrative control over the system.

- Admin Login – This feature provides secure access for administrators to manage and oversee the entire platform efficiently.
- Verify Service – Administrators can review and either approve or reject services submitted by students to maintain quality standards.
- Block / Unblock Service – Admins have the ability to restrict or restore services based on compliance or reported issues.
- View Users – This allows admins to monitor both student and client accounts to track activity and maintain platform integrity.
- Block / Unblock Users – Admins have the authority to restrict or reinstate user accounts to manage misuse or violations.
- View Service Rating – Admins can view feedback and ratings given to services to assess performance and user satisfaction.

View Complaint & Send Reply – Admins manage user complaints and provide timely responses to resolve concerns effectively.

3.4 User Story

A user story is a short description of a feature from the end user's perspective, used to guide development in Agile methodology. In SkillBridge: A Student Talent Marketplace Platform, examples include: “As a student, I want to list my services so that clients can view and request them,” “As a client, I want to browse and book services so that I can hire students easily,” and “As an admin, I want to verify services and manage users so that the platform remains secure and reliable.” These stories help ensure the system meets real user needs.

Table 3.4: User Story

User Story ID	**As a type of User**	**I want to**	**So that I can**
1	ADMIN	Login	Access the admin dashboard securely with valid credentials
		Verify services	Approve or reject student service listings to ensure quality & authenticity
		Block / Unblock users	Monitor user activity to keep the platform safe and trustworthy.
		View service ratings	Monitor service quality & ensure client satisfaction

User Story ID	As a type of User	I want to	So that I can
		View complaints & send reply	Solve issues raised by users/service
2	SERVICE PROVIDER (Student)	Register	Create an account to offer my services and display my skills
		Login	Access my profile and manage my requests securely.
		Manage works	Upload, update, or remove details of the services I offer
		View user requests	See service requests sent by clients for my listed skills
		Update request status	Update clients about the status of their requests
		Chat with user	Chat with clients to understand their requirements better
		Update price	Change my service price as needed.
		View feedback & rating	View client feedback to improve service

User Story ID	As a type of User	I want to	So that I can
		Send complaint & view reply	Send complaints to admin & get replies
3	USER (client)	Register	Create an account to access student service
		Login	Log in securely to manage your requests anytime
		View product & request	Browse student services & send services requests
		chat	Communicate directly with service providers for clarity on requirements
		View request status & make payment	Track request progress and make payments safely.
		Send feedback & rating	Share feedback to help improve the platform and services.
		Send complaint &view reply	Report any issues to admin and view their response

3.5 Product Backlog

The product backlog is a prioritized list of features and tasks that need to be implemented in the system. Each item includes an estimate of effort and its current development status.This backlog guides sprint planning and ensures that high-priority functionalities are delivered first.

Table 3.5: Product Backlog

ID	Name	Priority	Estimate(Hours)	Status
1	Admin Dashboard & Login	High	3	COMPLETED
2	Service Verification & Approval	High	8	COMPLETED
3	User Management (Block/Unblock)	Medium	6	COMPLETED
4	Complaint/Issue Resolution Interface	High	10	COMPLETED
5	View service ratings/quality monitoring	Medium	5	COMPLETED
6	Service provider Registration & Login	High	5	COMPLETED
7	Service management	High	12	COMPLETED

8	View client Requests &Details	High	7	COMPLETED
9	Update Request status	High	5	COMPLETED
10	Chat with client	High	15	COMPLETED
11	Client Registration & Login	High	5	COMPLETED
12	Browse/search services	High	10	COMPLETED
13	Send service request	High	5	COMPLETED
14	Chat with Service provider	High	15	COMPLETED
15	Track Request & secure payment	High	18	COMPLETED
16	Send Feedback & Rating	Medium	4	COMPLETED
17	Send complaint & View Admin Reply	Medium	6	COMPLETED

3.6 Project Plan

The project plan outlines the timeline and development phases for each user story, organized into sprints. It includes task names, start and end dates, estimated duration, and current status. This structured approach ensures timely delivery and alignment with Agile methodology.

Table 3.6: Project Plan

User Story ID	Task Name	Start Date	End Date	Days	Status
1,2	Sprint 1	06/08/2025	09/08/2025	4	COMPLETED
3,4	Sprint 1	10/08/2025	15/09/20205	6	COMPLETED
5,6	Sprint 2	16/08/2025	23/08/2025	7	COMPLETED
7,8	Sprint 2	25/08/2025	01/09/2025	8	COMPLETED
9,10	Sprint 3	02/09/2025	10/09/2025	9	COMPLETED
11,12	Sprint 3	13/09/2025	20/09/2025	8	COMPLETED
13,14	Sprint 4	22/09/2025	29/09/2025	7	COMPLETED
15,16,17	Sprint 4	30/10/2025	12/10/2025	13	COMPLETED

3.7 Sprint Backlog

The sprint backlog is a breakdown of tasks selected from the product backlog to be completed within each sprint. It includes task IDs, descriptions, assigned roles, estimated effort, and current status. This helps the development team stay focused and track progress during each sprint cycle.

Table 3.7: Sprint Backlog

Backlog Item	Status And Completion Date	Original Estimation in Hours	Day 1 hrs	Day 2 hrs	Day 3 hrs	Day 3 hrs	Day 5 hrs	Day 6 hrs	Day 7 hrs	Day 8 hrs	Day 9 hrs	Day 10 hrs
SPRINT 1												
Project setup	06/08/2025	3	1	1	1	0	0	0	0	0	0	0
Database Schema Design	10/08/2025	5	1	1	1	1	1	0	0	0	0	0
Module Planning & Routing	12/08/2025	4	1	1	1	1	0	0	0	0	0	0
SPRINT 2												
Admin Login & Dashboard	15/08/2025	6	1	1	1	1	1	1	0	0	0	0
Service & User Verification	18/08/2025	5	1	1	1	1	1	0	0	0	0	0
Complaint & Rating Management	22/08/25	6	1	1	1	1	1	1	0	0	0	0
SPRINT 3												
Student Registration & Portfolio	27/08/2025	6	1	1	1	1	1	1	0	0	0	0

Request Management & Pricing	01/09/2025	7	1	1	1	1	1	1	1	0	0	0
Chat & Feedback Integration	07/09/2025	6	1	1	1	1	1	1	0	0	0	0
SPRINT 4												
Client Registration	13/09/2025	6	1	1	1	1	1	1	0	0	0	0
Booking, Payment & Status View	22/09/2025	7	1	1	1	1	1	1	1	0	0	0
Feedback & Complaint Handling	08/10/2025	5	1	1	1	1	1	0	0	0	0	0
TOTAL		66	12	12	12	11	10	7	2			

3.8 Database Design

The database design defines how data is structured, stored, and related within the SkillBridge platform. Using MySQL, the system is organized into tables such as users, service providers, works, requests, bookings, feedback, and notifications, each tailored to handle specific functionalities. The design ensures efficient data retrieval, scalability, and consistency, supporting core features like service registration, work management, client requests, secure authentication, chat communication, booking management, and feedback tracking.

Table 3.8: Database Design

Collection	Attributes	Purpose
User	• userId (PK) • username • email • password • role (admin / service_provider / client) • status	Stores details of all registered users including admins, service providers, and clients
Service	• serviceId (PK) • providerId (FK → users) • service_name • category • description • price • status	Stores information about services offered by student providers
Requests	• requestId (PK) • userId (FK → users) • serviceId (FK → services) • request_date • status • amount • payment_status	Stores service requests made by users and their status
Chats	• chatId (PK) • senderId (FK → users) • receiverId (FK → users) • message • sent_time	Stores messages exchanged between users and service providers

Feedback	• feedbackId (PK) • userId (FK → users) • serviceId (FK → services) • rating • comment • feedback_date	Stores feedback and ratings given by users for completed services
Complaints	• complaintId (PK) • userId (FK → users) • message • reply • submitted_date • status	Stores complaints submitted by users and replies from admin
Admin_reports	• reportId (PK) • generated_by (FK → users) • report_type • generated_date • details	Stores system-generated or admin-generated reports for monitoring

Chapter 4. Results and Discussions

SkillBridge: A Student Talent Marketplace Platform was successfully developed with three main modules: **Service Provider (Student), User (Client), and Admin**. Students can register, create profiles, list services, manage requests, set pricing, and communicate with clients. Clients can register, browse services, submit requests, track progress, make payments, and provide feedback. Admins can verify services, manage users, monitor activities, and handle complaints.

The system effectively addressed key problems of informal talent marketplaces, such as unorganized service listings, lack of secure payments, poor communication, and difficulty in verifying user credibility. With real-time request tracking, integrated chat, feedback and rating systems, and secure payment processing, SkillBridge improves convenience for clients, efficiency for students, and oversight for admins.

Overall, the platform met its objectives by replacing scattered and unreliable methods with a centralized, structured, and user-friendly digital marketplace, enhancing opportunities for students to showcase skills and earn income while providing clients with reliable and verified talent.

4.1 Results

This section highlights the main working pages of SkillBridge: A Student Talent Marketplace Platform that demonstrate how the implemented modules function in practice. The results are presented with screenshots of important interfaces such as student registration and login, service listing and management, client dashboard, request tracking, payment processing, chat interface, and the admin panel**.**

These results validate that the proposed system meets its objectives by enabling students to easily manage their services and requests, clients to browse and book services efficiently, and administrators to effectively monitor platform activities and handle complaints. Overall, the system provides a reliable and user-friendly solution for connecting students with clients and managing a structured digital talent marketplace.

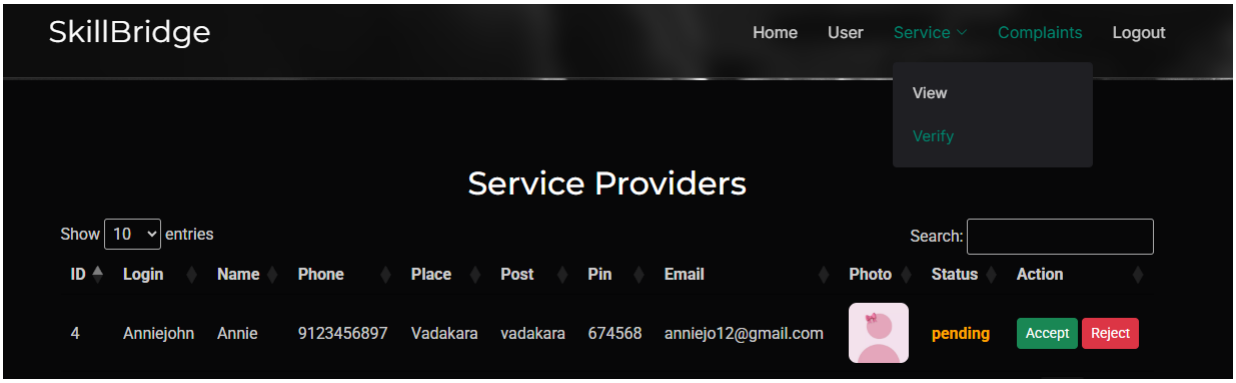


Figure 4.1: Admin verification panel

Figure 4.1 The Admin Verification Panel is ensures the authenticity of users and service providers by validating their details before approval. It maintains the platform's integrity and promotes a secure, trustworthy environment for all participants.

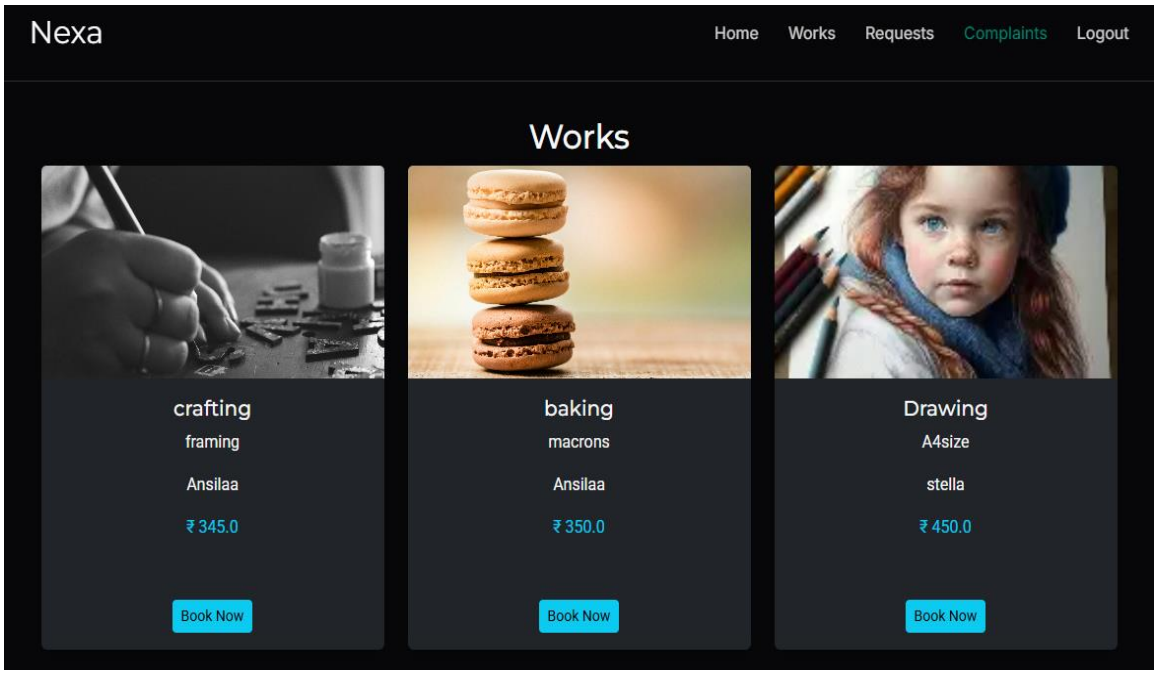


Figure 4.2: View User Works

Figure 4.2 This feature lets users see the works posted by students. They can browse through the projects, check details, and easily find the skills and contributions of each student. It makes viewing student work simple and organized.

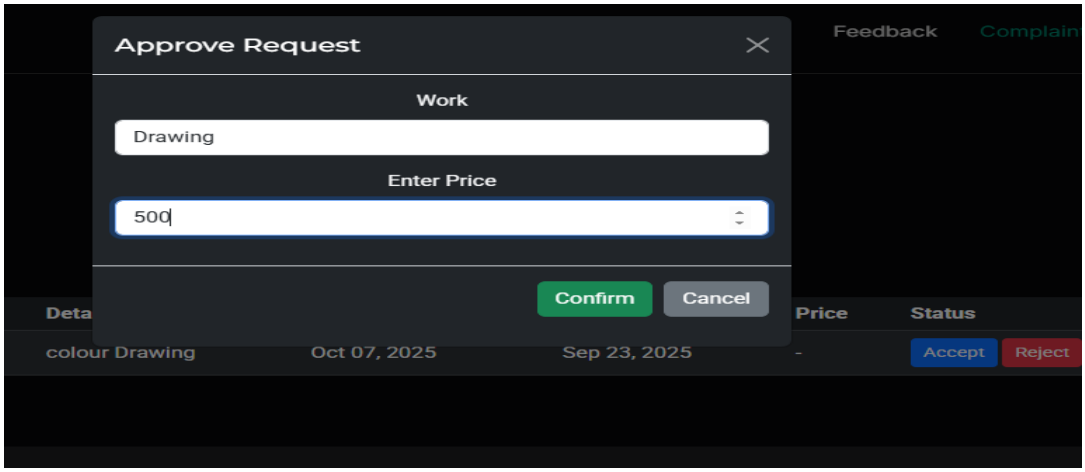


Figure 4.3: Confirm Request

Figure 4.3 This Approve Request modal is where the service provider enters the price for the requested Drawing before clicking Confirm to formalize the accepted service agreement.

Chapter 5. Conclusion and Future Scope

The SkillBridge: A Student Talent Marketplace Platform effectively addresses the challenges faced by students in showcasing and monetizing their skills. By offering a centralized and structured platform, it enables students to connect with potential clients within and outside their campus in a secure and organized manner. The system ensures smooth communication, safe payments, and transparency through features like real-time chat, request tracking, and admin verification. SkillBridge not only helps students gain practical experience and income but also builds client trust by ensuring the quality and authenticity of services. Overall, the platform successfully achieves its goal of empowering student freelancers and simplifying service access for clients.

There is significant potential to enhance the SkillBridge platform in future versions. A dedicated mobile application can be developed to make the platform more accessible and user-friendly. AI-based matching between clients and service providers can improve efficiency and user experience. Features like automated review moderation, digital certification for verified talents, and support for multiple payment gateways can increase platform credibility. Integration with university portals, internship listings, or even placement services can also expand the platform's reach and utility. With ongoing improvements, SkillBridge can evolve into a comprehensive career-launching ecosystem for students.

References

[1] D. Nguyen and S. Patel, “Campus-based Freelance Platforms: Bridging Skill and Opportunity,” *Journal of Innovation in Digital Economy*, vol. 3, no. 2, pp. 23–30, 2021.

[2] Microsoft Copilot, AI Companion for Technical and Creative Assistance. [Online]. Available: https://copilot.microsoft.com/.

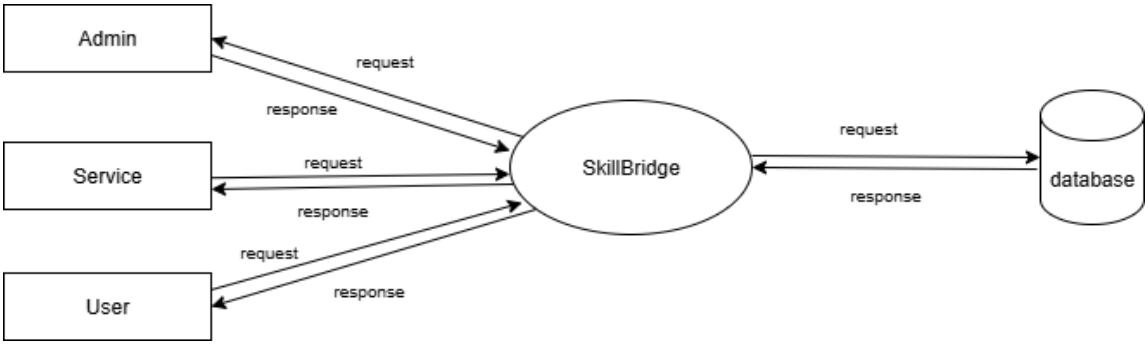
[3] W3Schools, “Web Development Tutorials for HTML, CSS, and JavaScript,” [Online]. Available: https://www.w3schools.com/.

[4] Oracle, “MySQL 8.0 Reference Manual,” https://dev.mysql.com/doc/.

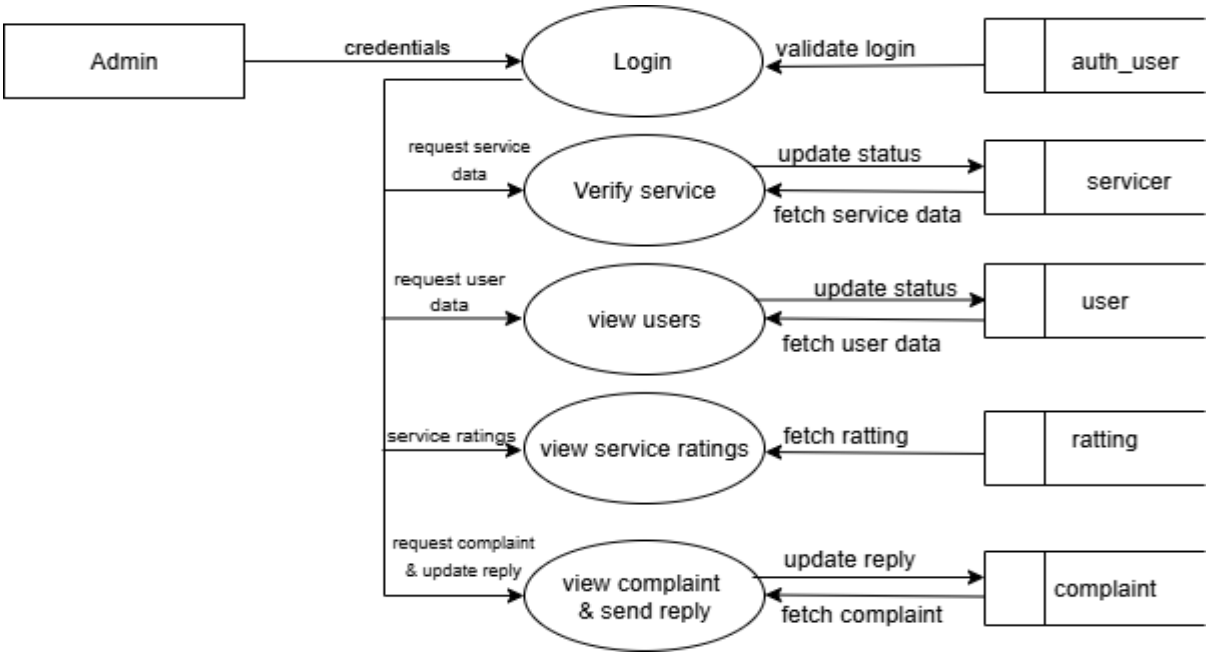
Appendix

Appendix A Data Flow Diagram

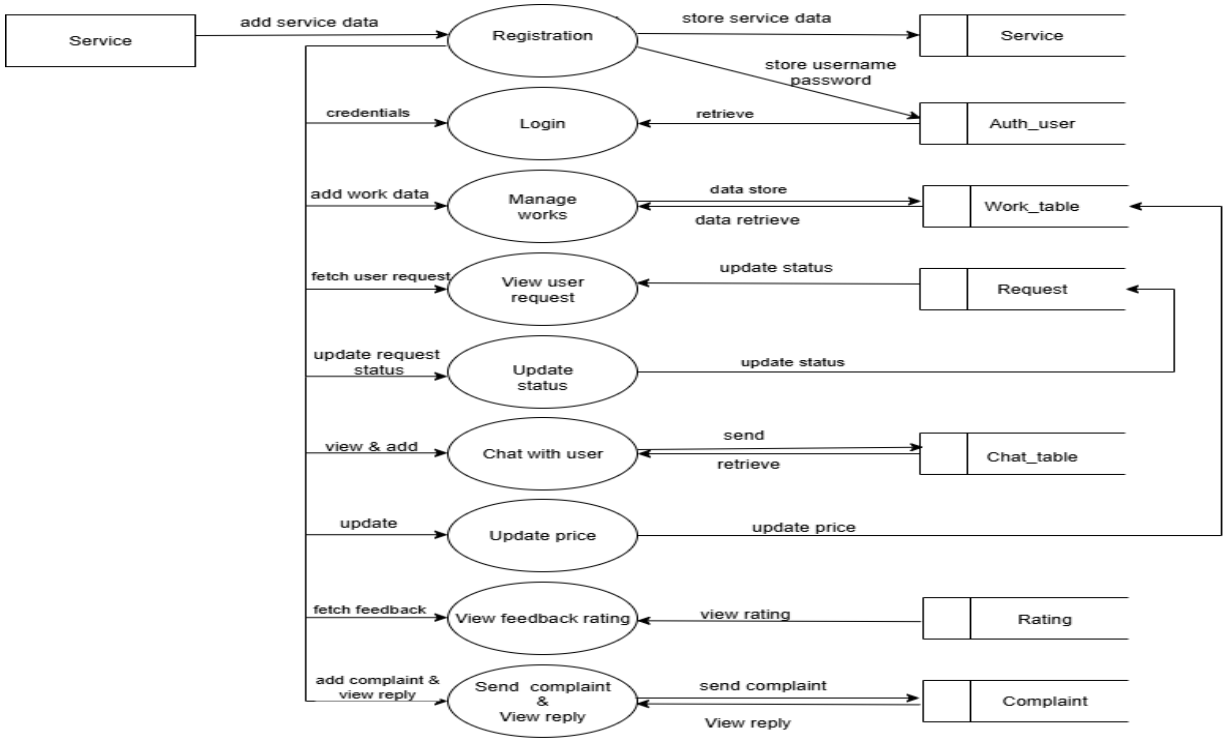
DFD Level 0



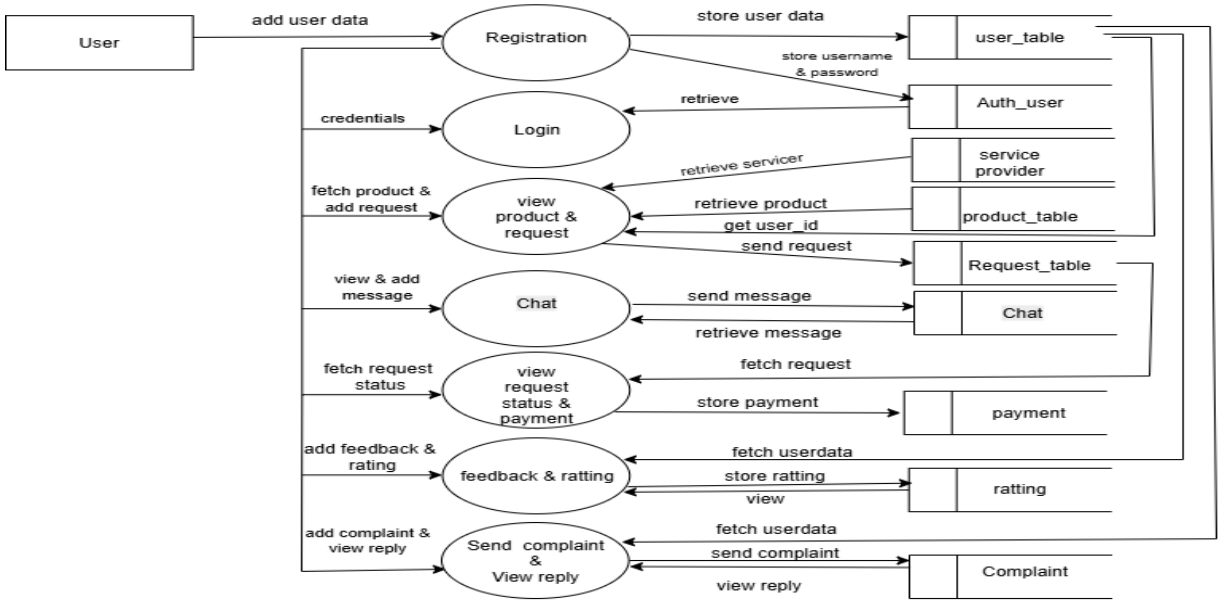
DFD Level 1



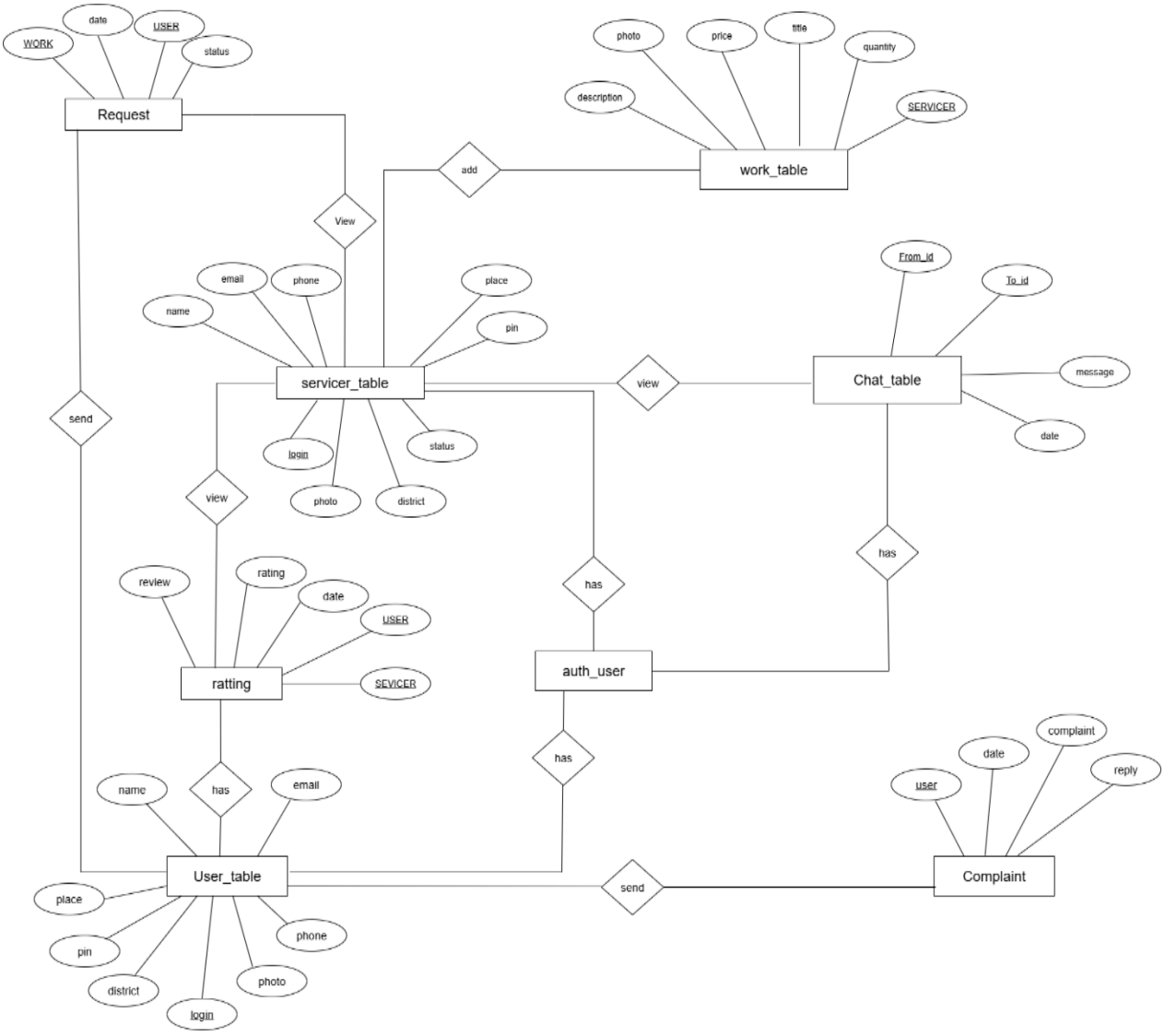
DFD Level 2



DFD Level 3



Appendix B ER Diagram



Appendix C Source Code

```
from django.contrib import messages
from django.http.response import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.contrib.auth import authenticate, login, logout
from django.shortcuts import render,redirect
from django.contrib.auth.models import Group,User
from .models import *
from django.contrib.auth.hashers import check_password, make_password
from datetime  import datetime

from django.contrib.auth.decorators import login_required
def login_page(request):
    logout(request)
    return render(request,"index.html")
def loginpost(request):
    username = request.POST["username"]
    password = request.POST["password"]
user = authenticate(request, username=username, password=password)
    print(user)
    if user is not None:
        print("hhh")
        if user.groups.filter(name="Admin").exists():
            print("jajaja")
            login(request, user)
            return redirect('/myapp/admin_home/')
        if user.groups.filter(name="Service").exists():
            print("jajaja")
            ob=Service_Table.objects.get(LOGIN__id=user.id)
            if ob.status=='service':
                login(request, user)
                return redirect('/myapp/service_home/')
        if user.groups.filter(name="User").exists():
            print("jajaja")
            login(request, user)
            return redirect('/myapp/user_home/')
    else:
        messages.warning(request, "Invalid username or password")
        return redirect('/myapp/')
    messages.warning(request, "Invalid username or password")
    return redirect('/myapp/')
def logouta(request):
    logout(request)
    return redirect('/myapp/')
```

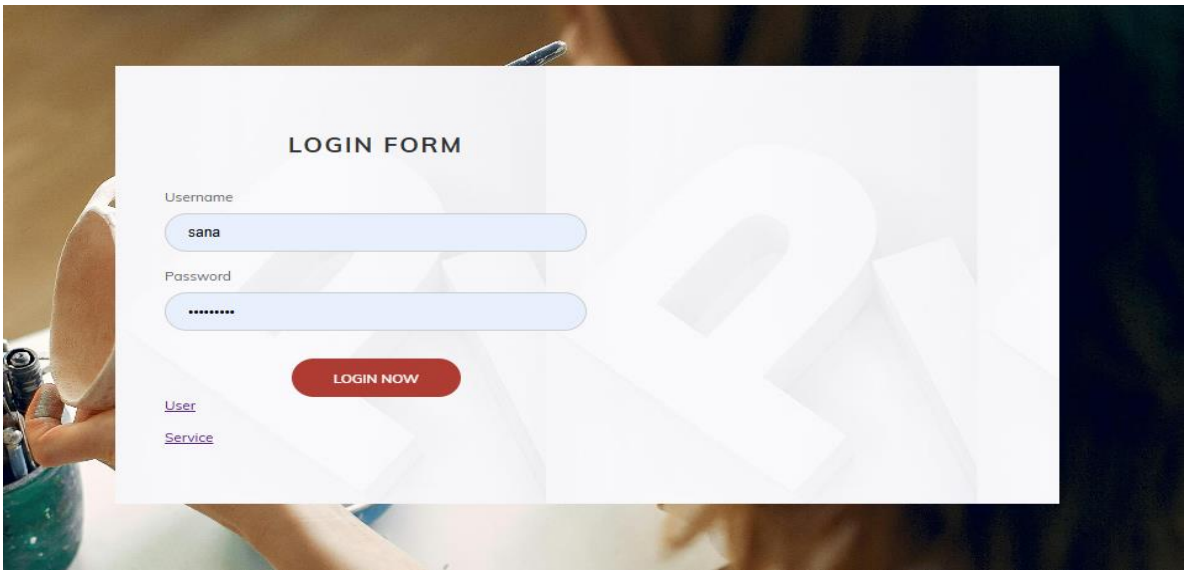
```
def registration_service(request):
    if request.method == "POST":
        name = request.POST["name"]
        phoneno = request.POST["phoneno"]
        place = request.POST["place"]
        post = request.POST["post"]
        pin = request.POST["pin"]
        email = request.POST["email"]
        password = request.POST["password"]
        username=request.POST['uname']
        photo = request.FILES["photo"]
        user = User.objects.create(username=username, password=make_password(password),
email=email, first_name=name)
        user.save()
        user.groups.add(Group.objects.get(name="Service"))
        Service_Table.objects.create(
            LOGIN=user,
            name=name,
            phoneno=phoneno,
            place=place,
            post=post,
            pin=pin,
            email=email,
            photo=photo,
        )
        messages.success(request,"Success")
        return redirect("/myapp/")
    return render(request,"registration_service.html")
@login_required(login_url='/myapp/')

def user_register(request):
    if request.method == "POST":
        username = request.POST["username"]
        password = request.POST["password"]
        name = request.POST["name"]
        phoneno = request.POST["phoneno"]
        place = request.POST["place"]
        post = request.POST["post"]
        pin = request.POST["pin"]
        email = request.POST["email"]
        photo = request.FILES["photo"]
        user = User.objects.create(username=username, password=make_password(password),
email=email, first_name=name)
        user.save()
        user.groups.add(Group.objects.get(name="User"))
# Create user profile
```

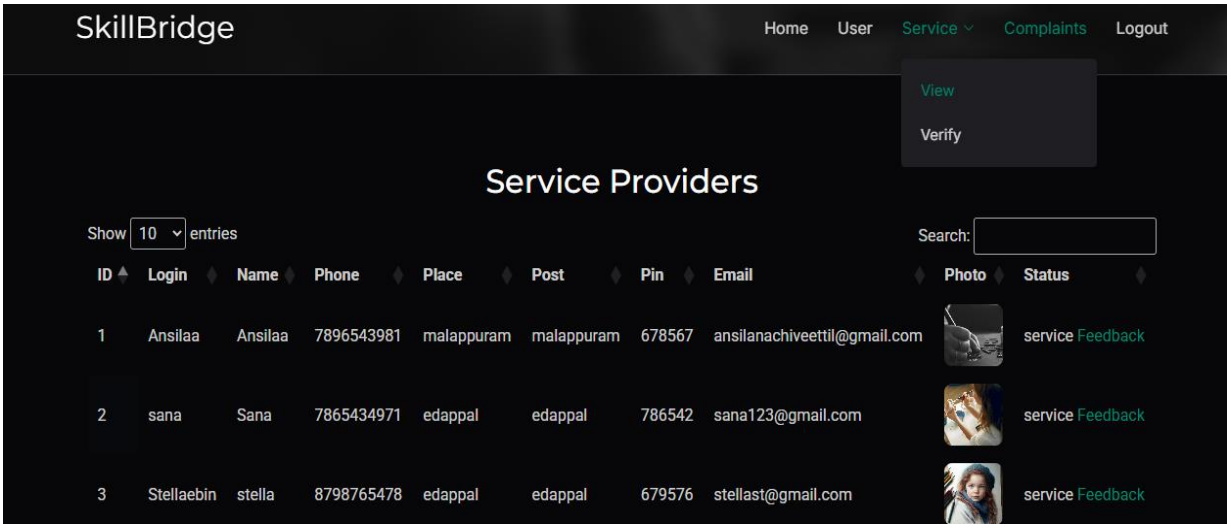
```
        User_Table.objects.create(
            LOGIN=user,
            name=name,
            phoneno=phoneno,
            place=place,
            post=post,
            pin=pin,
            email=email,
            photo=photo,
            status="user"
        )
        messages.success(request, "Success")
        return redirect("/myapp/")  # after registration redirect to login page
    return render(request, "user_reg.html")
@login_required(login_url='/myapp/')
# Chat Service
@login_required(login_url='/myapp/')
def chat_view_service(request,id):
    try:
        request.session["userid"] = id
        qry=User_Table.objects.get(LOGIN__id=id)
        request.session["new"] = id
        return render(request, "Chat.html",{'photo': qry.photo.url, 'name': qry.name, 'toid': id})
    except:
        request.session["userid"] = id
        qry=Service_Table.objects.get(LOGIN__id=id)
        request.session["new"] = id
        return render(request, "Chat.html",{'photo': qry.photo.url, 'name': qry.name, 'toid': id})

@login_required(login_url='/myapp/')
def chat_send(request, msg):
    lid = request.user.id
    toid = request.session["userid"]
    message = msg
    import datetime
    d = datetime.datetime.now().date()
    chatobt = Chat_Table()
    chatobt.message = message
    chatobt.TO_id = toid
    chatobt.FROM_id = lid
    chatobt.date = d
    chatobt.save()
    return JsonResponse({"status": "ok"})
```

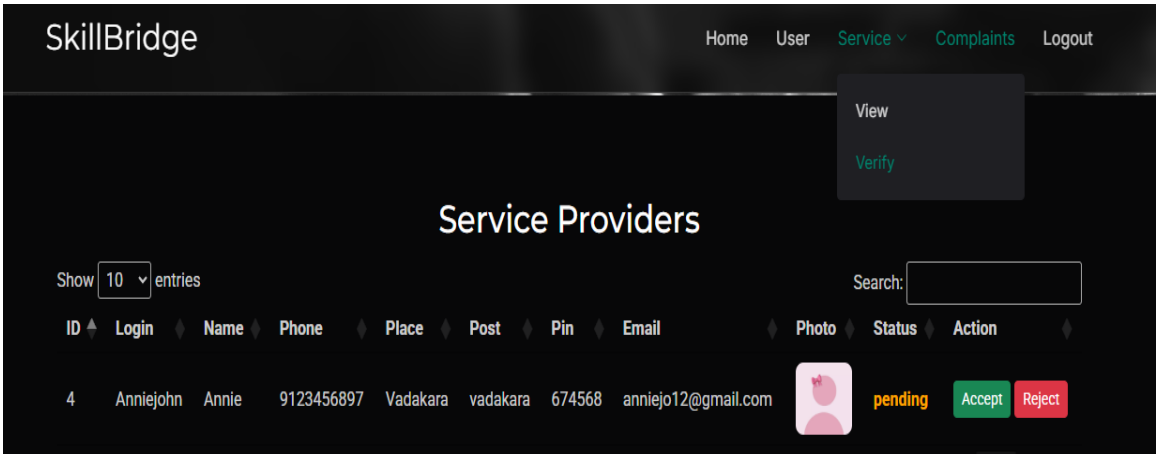
Appendix D Screenshot



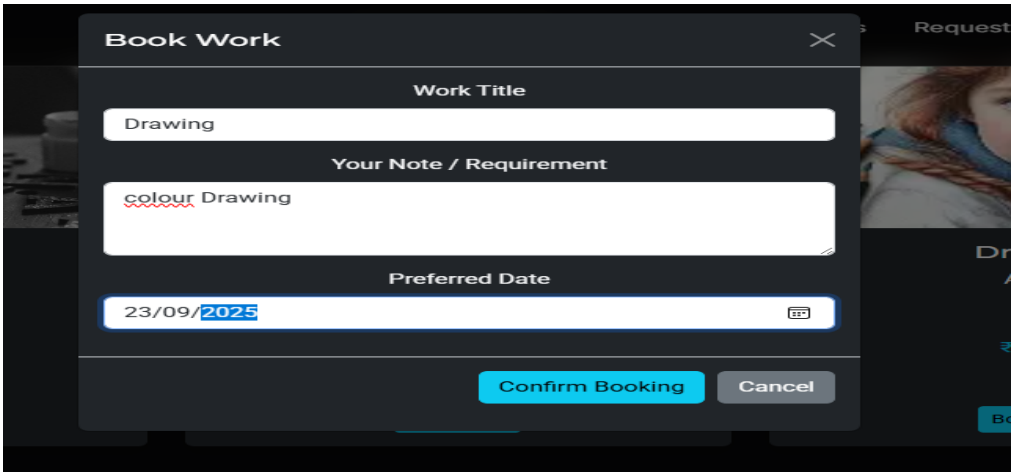
ADMIN ACCESS PANEL



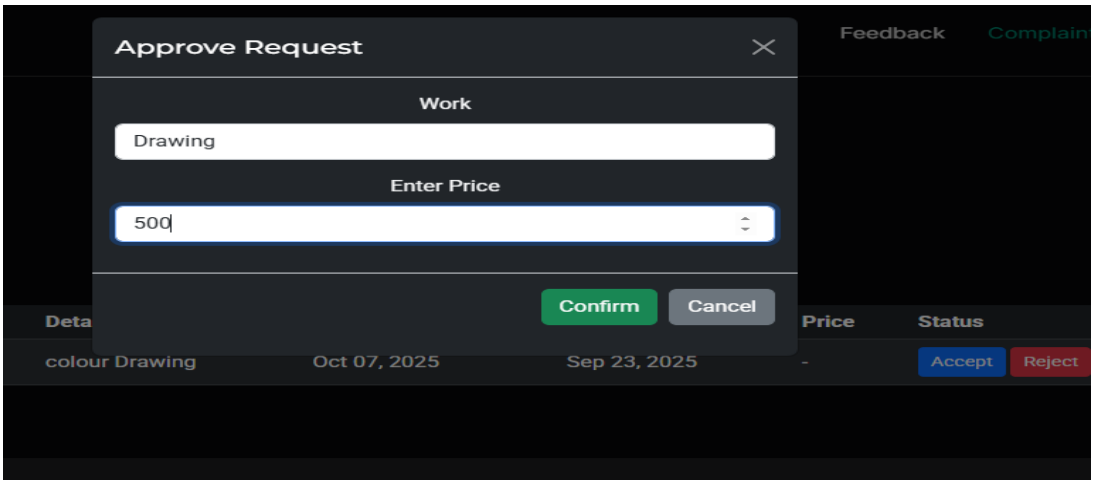
SERVICE PROVIDER DASHBOARD



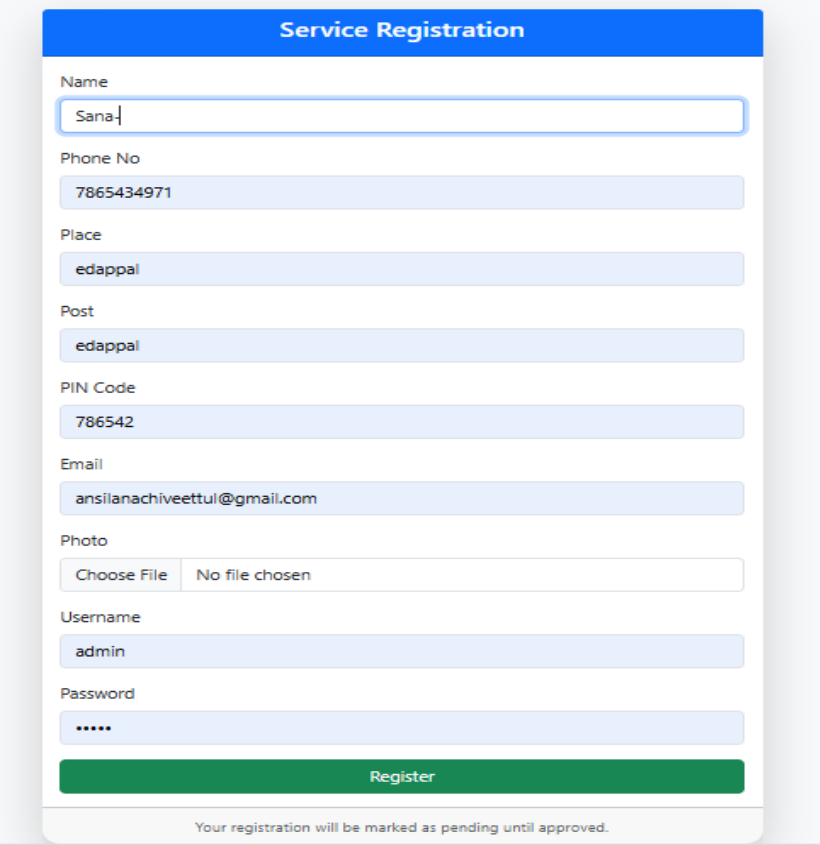
ADMIN VERIFICATION PANEL



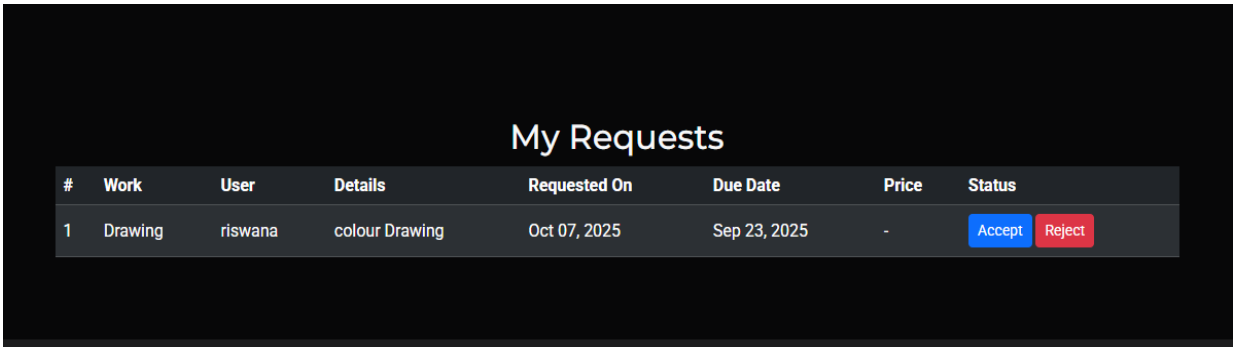
USER BOOKING PANEL



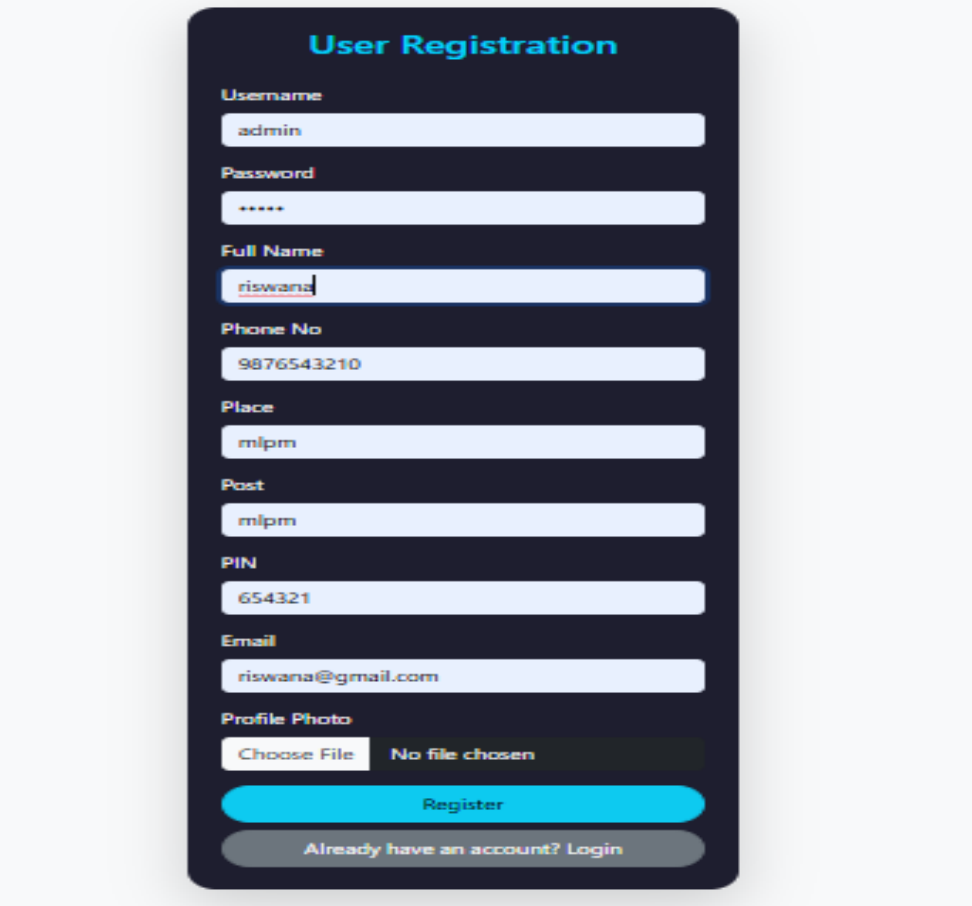
SERVICE APPROVAL



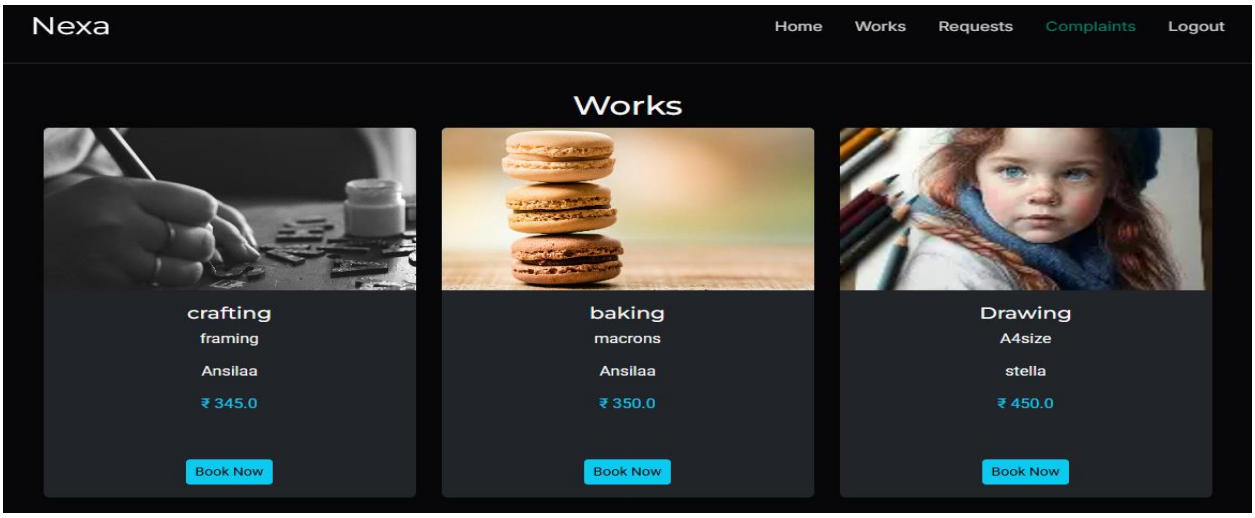
SERVICE REGISTRATION



SERVICE REQUEST VIEW



USER REGISTRATION



VIEW USER WORKS