

# **MARKETPLACE FOR EDUCATIONAL SERVICES**

**‘EduMart’**

**PROJECT REPORT**

**PHASE II**

**MASTER OF COMPUTER APPLICATIONS**

**(4<sup>th</sup> Semester)**

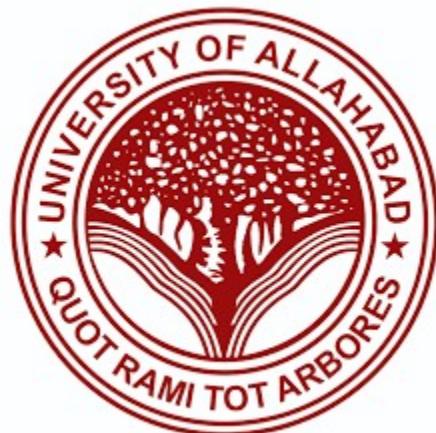
**SUBMITTED BY**

**DINESH SINGH**

**Batch-2023-2025**

**Enrollment No.-U2046035**

**PROJECT GUIDE- Ms. SUNITA TRIPATHI**



**Centre of Computer Education & Training**

**Institute of Professional Studies**

**University of Allahabad, Prayagraj**

**Uttar Pradesh – 211002**

## **ACKNOWLEDGEMENT**

I would like to take this opportunity to express my heartfelt gratitude to everyone who has supported and guided me throughout the completion of this project report. Its successful completion is undoubtedly the result of the invaluable support, encouragement, and contributions of many individuals.

I am deeply grateful to **Professor Ashish Khare**, Coordinator, Centre of Computer Education and Training, University of Allahabad, for his valuable guidance and for providing the necessary resources and academic environment to carry out this project.

First and foremost, I extend my sincere thanks to my project supervisor, **Ms. Sunita Tripathi** (Centre of Computer Education and Training, Institute of Professional Studies, University of Allahabad), for her continuous guidance, insightful suggestions, and for fostering a motivating and encouraging environment. Her unwavering support has been instrumental in shaping and completing this project.

I would also like to express my gratitude to **Dr. Sarika Yadav** and **Dr. Farha** for their support and encouragement throughout the course of the project.

My sincere thanks to all the teachers who have extended their help and guidance during this journey.

Lastly, I am deeply thankful to my family and friends for their unwavering support, motivation, and belief in me. Their constant encouragement has been a source of strength throughout the duration of this project.

## **DECLARATION**

I, Dinesh Singh, hereby declare that the project titled "**Marketplace for Educational Services (EduMart)**", submitted to the **Centre of Computer Education & Training, Institute of Professional Studies, University of Allahabad**, is my original work, completed under the guidance of **Ms. Sunita Tripathi**. This project report has not been previously submitted to any other institution for any degree or diploma.

The information presented in this report is accurate to the best of my knowledge. All references and sources have been properly acknowledged, and no part of this report contains plagiarized content. I understand that any instance of plagiarism, including the use of uncredited work from others, may result in severe academic penalties and the potential rejection of this project.

**Date:**

**Place:** CCET, IPS, UoA, Prayagraj

**Dinesh Singh**

(Enrollment No.- U2046035)

## CERTIFICATE

This is to certify that **Dinesh Singh**, a student of the **Master of Computer Applications (Semester – 4<sup>th</sup>)** program at the Centre of Computer Education and Training, Institute of Professional Studies, University of Allahabad, has completed his work on the project entitled **“Marketplace for Educational Services (EduMart)”** under my guidance during the academic session 2024-25.

He has taken proper care and demonstrated the utmost sincerity in completing this project. I certify that his work adheres to the guidelines, is original, and has never been submitted elsewhere for the same purpose.

**Date:**

**Place:** CCET, IPS, UoA, Prayagraj

**Ms. Sunita Tripathi**

(Project Supervisor)

## TABLE OF CONTENTS

<b>Sr. No.</b>	<b>Contents</b>	<b>Page No.</b>
	Acknowledgement	(i)
	Declaration	(ii)
	Certificate	(iii)
	Synopsis	(iv)
1	Introduction	1-3
	1.1 Problem Definition	
	1.2 Motivation	
2	Objective	4
3	Goals	5
4	Requirement Analysis & Specifications	6-9
	4.1 Functional Requirements	
	4.2 Non-Functional Requirements	
	4.3 Technical Requirements	
	4.4 Software & Hardware Requirements	
5	System Design (DFD)	10-14
6	Development & Testing	15
7	Modules	16
8	Test Cases	17-19
9	Project Outputs	20-29
10	Codes	30-101
11	Applications	102
12	Conclusion	103
13	Future Scope	104
14	Milestones	105
15	Meetings with the Supervisor	106
16	References	107
17	Plagiarism Report	108

# **SYNOPSIS**

# **MARKETPLACE FOR EDUCATIONAL SERVICES**

**‘EduMart’**

## **PROJECT SYNOPSIS**

**PHASE II**

**MASTER OF COMPUTER APPLICATIONS**

**(4<sup>th</sup> Semester)**

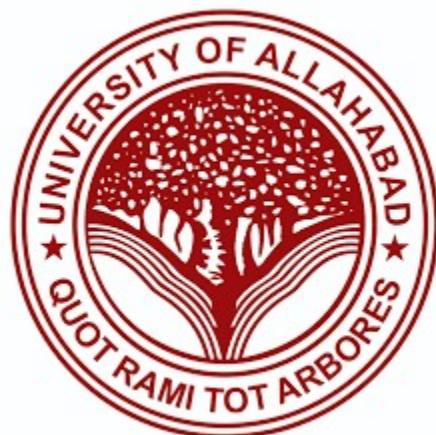
**SUBMITTED BY**

**DINESH SINGH**

**Batch-2023-2025**

**Enrollment No.-U2046035**

**PROJECT GUIDE- Ms. SUNITA TRIPATHI**



**Centre of Computer Education & Training**

**Institute of Professional Studies**

**University of Allahabad, Prayagraj**

**Uttar Pradesh – 211002**

## **TABLE OF CONTENTS**

<b>Sr. No.</b>	<b>Contents</b>	<b>Page No.</b>
1	Introduction	1-3
	1.1    Problem Definition	
	1.2    Motivation	
2	Objective	4
3	Goals	5
4	Requirement Analysis & Specification	6-9
	4.1    Functional Requirements	
	4.2    Non-Functional Requirements	
	4.3    Technical Requirements	
	4.4    Software & Hardware Requirements	
5	System Design (DFD)	10-14
6	Development & Testing	15
7	Milestones	16
8	Meetings with the Supervisor	17
9	References	18

## 1. INTRODUCTION

**EduMart** will be developed to connect individuals who will seek educational services with skilled professionals who will offer them. The objective of this platform will be to enable clients to discover, book, and procure a wide range of services such as tutorials, graphic designing, and application development. At the same time, it will empower service providers to showcase their expertise and manage bookings efficiently.

This project will focus on creating a user-friendly and secure web platform that will facilitate seamless interactions between service providers and clients. **EduMart** will utilize modern technologies to ensure scalability, data security, and smooth operations. By introducing features such as advanced search, real-time booking, and a review system, EduMart will aim to simplify the process of accessing and delivering educational services, making it a reliable hub for educational needs.

**Project Title:** ‘EduMart’ (Marketplace for Educational Services)

**Category:** Web Development

### 1.1. PROBLEM DEFINITION

In the current scenario, individuals seeking educational or skill-based services face challenges in finding reliable service providers. Traditional methods of searching through advertisements or referrals are often time-consuming and lack transparency regarding the quality and credibility of the services offered. Similarly, skilled professionals face difficulties in reaching their target audience and showcasing their expertise effectively.

Existing platforms may lack specialized focus on educational services, leading to a cluttered experience with irrelevant offerings. Moreover, the absence of streamlined processes for booking, secure payments, and real-time communication results in inefficiencies and a lack of trust between service providers and clients.

For instance, clients might struggle to compare multiple service providers based on their skills, experience, and pricing. On the other hand, service providers may not have the tools to highlight their unique capabilities, manage client interactions, or receive constructive feedback. This gap creates a pressing need for a dedicated, well-organized platform that caters exclusively to educational services, ensuring ease of use, transparency, and mutual trust.

**EduMart** will aim to address these issues by creating a marketplace that will simplify the process of discovering, booking, and availing educational services while empowering providers to showcase their skills and build their reputation in a secure and transparent environment.

## 1.2. MOTIVATION

**EduMart** will be driven by the need to bridge the existing gap between clients seeking educational services and skilled professionals offering them. In the traditional landscape, individuals often face challenges in finding reliable service providers, especially in the educational domain, where quality, transparency, and efficiency are crucial. Existing platforms are often cluttered, and the search for educational services becomes a time-consuming task without a centralized and organized approach. Moreover, service providers struggle to showcase their skills and connect with clients effectively.

The motivation behind **EduMart** is to create a specialized marketplace that focuses exclusively on educational and skill-based services, ensuring that both clients and providers have a streamlined, secure, and transparent platform to interact. By introducing advanced features such as real-time bookings, secure payments, review systems, and categorized listings, **EduMart** will aim to eliminate the inefficiencies of traditional methods of finding educational services, making the entire process smoother, more trustworthy, and accessible for everyone involved.

**EduMart** will strive to address the challenges faced in the educational services space. It will aim to remove the obstacles related to service discovery, booking processes, and transparent interactions, empowering clients and service providers alike. The platform's goal will be to simplify the search, comparison, and procurement of educational services while helping professionals build their reputation and establish themselves in a secure, user-friendly environment.

## 2. OBJECTIVE

**EduMart** will aim to create a secure, efficient, and user-friendly platform that will connect clients with skilled professionals offering educational services. It will ensure seamless accessibility for discovering, booking, and availing services while empowering providers to showcase their expertise and manage bookings effectively. With advanced search filters, real-time availability, secure payments, and a review system, **EduMart** will streamline processes, build trust, and foster transparency. Prioritizing security and scalability, it will safeguard user data and accommodate growth without compromising performance, offering a customized and reliable marketplace for both clients and service providers.

### 3. GOALS

The main goals of this project are:

1. **Centralized Marketplace:** Build a platform connecting clients with educational and skill-based service providers.
2. **Enhanced User Experience:** Create a user-friendly interface for seamless navigation and bookings.
3. **Trust and Transparency:** Implement reviews and ratings to ensure accountability and quality services.
4. **Empower Providers:** Enable professionals to showcase expertise and manage bookings efficiently.
5. **Streamlined Processes:** Simplify service discovery and bookings with advanced tools and real-time updates.
6. **Security:** Protect user data and payments with secure authentication and encryption.
7. **Customization:** Offer personalized profiles and tailored listings for better user relevance.
8. **Scalability:** Design the platform to handle growth without compromising performance.
9. **Real-Time Communication:** Integrate messaging for seamless client-provider interactions.
10. **Continuous Improvement:** Monitor feedback and update features to enhance functionality.

## 4. REQUIREMENT ANALYSIS & SPECIFICATIONS

### 4.1. Functional Requirements

- **User Registration and Authentication:** Users (both clients and service providers) must be able to register, log in, and manage their profiles securely using modern authentication mechanisms like email/password or social media login options.
- **Service Listings:** Service providers should be able to list their services with detailed descriptions, pricing, and availability. Services should be categorized into different educational subdomains such as tutorials, design, development, etc.
- **Advanced Search and Filters:** Clients should be able to search for services using multiple filters like price range, location, ratings, and service category to find the most relevant options.
- **Real-Time Booking:** Clients should be able to book services based on the real-time availability of service providers. The system should automatically update bookings and sync with service providers' schedules.
- **Payment Gateway Integration:** A secure payment gateway must be integrated to handle transactions, ensuring that both clients and service providers are paid promptly and securely.
- **Review and Rating System:** Clients should be able to leave feedback, ratings, and reviews for the services they availed, helping build trust and transparency on the platform.
- **Messaging System:** A direct messaging system for clients and service providers to communicate about services, bookings, or any clarifications.
- **Django Admin Panel:** The Django admin (built-in interface for managing database models, allowing CRUD operations and administrative tasks through a user-friendly web interface) must be able to manage user accounts, approve or reject service listings, monitor transactions, resolve disputes, and ensure compliance with platform policies.
- **User Profiles:** Both clients and service providers should have profiles to track orders, manage favourites, update their listings, and showcase testimonials or portfolios (for service providers).

## 4.2. Non-Functional Requirements

- **Performance:** The platform must handle high traffic efficiently, providing quick response times when browsing, booking, and making payments.
- **Usability:** The platform should have an intuitive and easy-to-navigate interface that caters to both tech-savvy users and beginners. It should offer a responsive design for users on desktop, tablet, and mobile devices.
- **Scalability:** The system should be designed to scale as the number of users and services grows without compromising on performance. This includes handling increased traffic, service listings, and data storage.
- **Security:** Ensure robust security measures for protecting sensitive user data, such as encrypted passwords, secure payment processing, and protection against unauthorized access. Compliance with data protection laws (e.g., GDPR) should also be a priority.
- **Compatibility:** The platform should be compatible with various web browsers (Brave, Chrome, Firefox, Safari, etc.) and operate smoothly across different operating systems (Windows, macOS, Linux).
- **Maintainability:** The system should be built with easy-to-maintain code, allowing for simple updates, bug fixes, and future feature enhancements. Comprehensive documentation should be available for developers.
- **Backup and Data Recovery:** Regular backups of user data and transaction logs should be in place, ensuring that data can be restored in the event of a failure or data loss incident.
- **Accessibility:** Ensure the platform meets accessibility standards, providing features such as text-to-speech for visually impaired users and easy navigation for users with disabilities.

### 4.3. Technical Requirements

#### 1. Frontend Technologies

- **HTML:** Creating Structure and markup for webpages.
- **CSS:** Styling and layout using Custom CSS.
- **JavaScript:** Client-side scripting for interactive features and game logic.
- **Bootstrap:** Framework for more prebuilt features for defining interactive UI.

#### 2. Backend Technologies

- **Django:** Framework for building the backend, handling routing, and interacting with the database.
- **Database:** SQLite for storing user data, game data, and performance metrics.
- **ORM:** Django's ORM (Object Relational Mapping) for database interactions.

## 4.4. SOFTWARE & HARDWARE REQUIREMENTS

### **Hardware Required:**

<b><u>Name of Components</u></b>	<b><u>Specifications</u></b>
Processor	: Intel/AMD/Ryzen
RAM	: 4GB or More
Speed	: 1.00 GHz
System Type	: 64-bit operating system

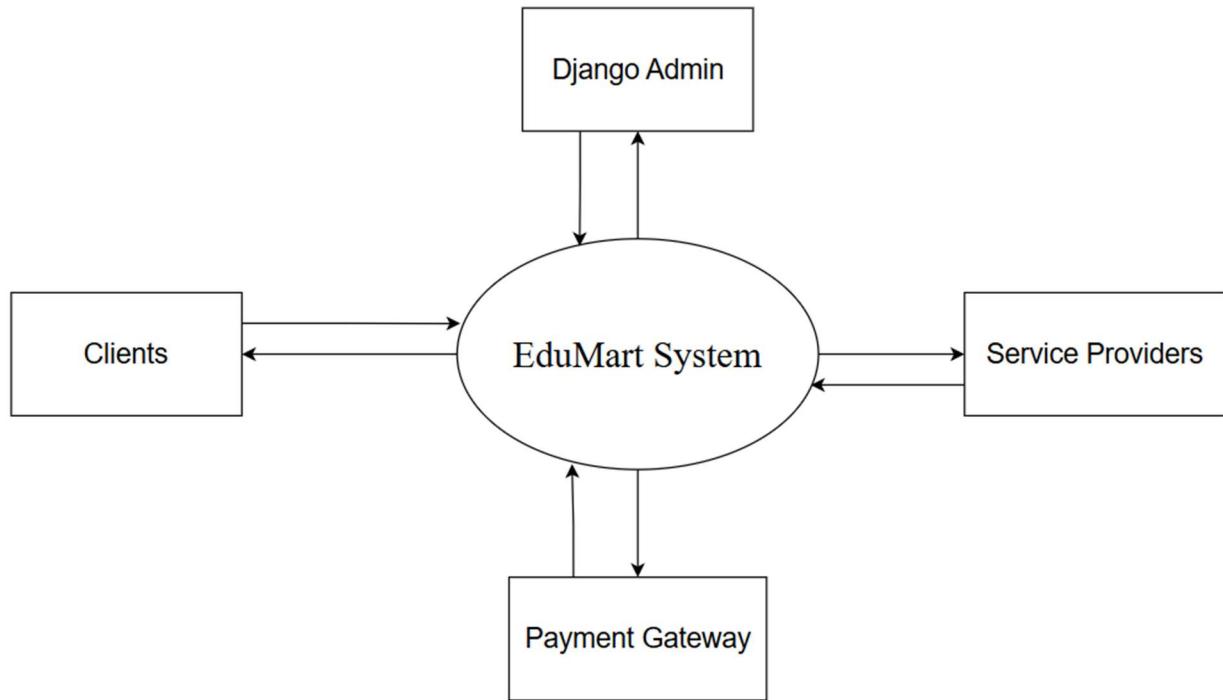
### **Software Required:**

<b><u>Name of Components</u></b>	<b><u>Specifications</u></b>
Operating System	: Windows/Mac/Linux
Frontend	: HTML, CSS, JavaScript
Framework	: Django, Bootstrap
Database	: SQLite
IDE	: Visual Studio Code
Browser	: Chrome/Mozilla/Edge/Brave

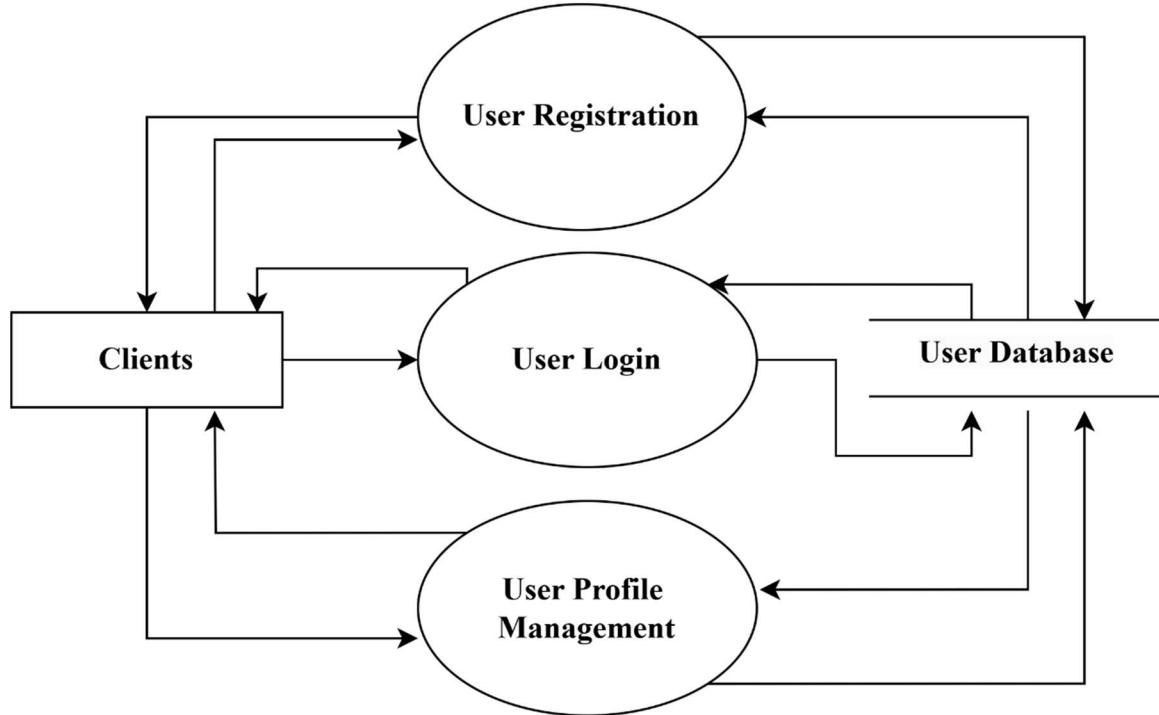
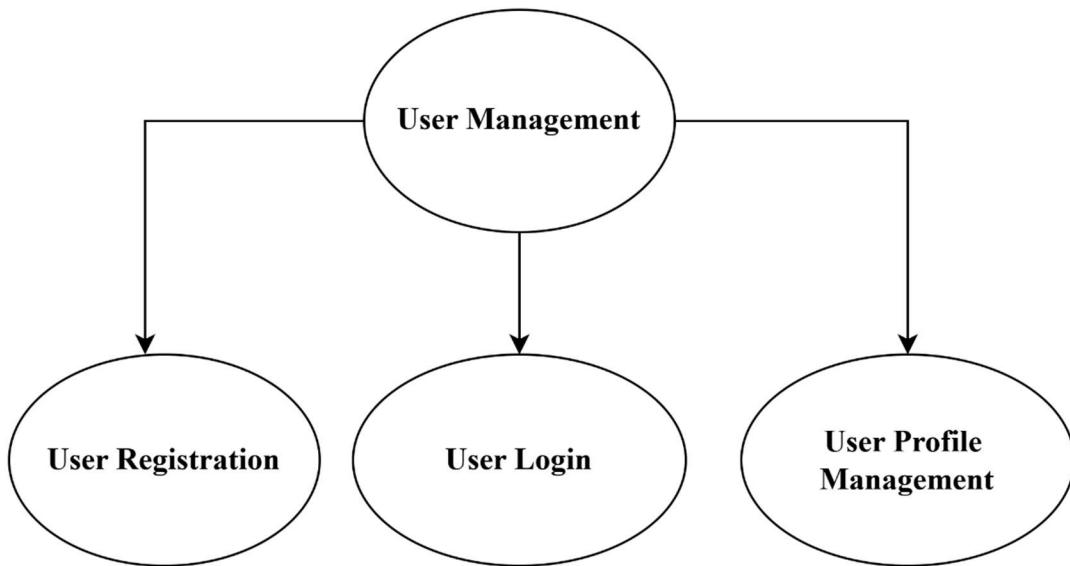
## 5. SYSTEM DESIGN

Detailed System design of **EduMart** using Data Flow Diagram (DFD):

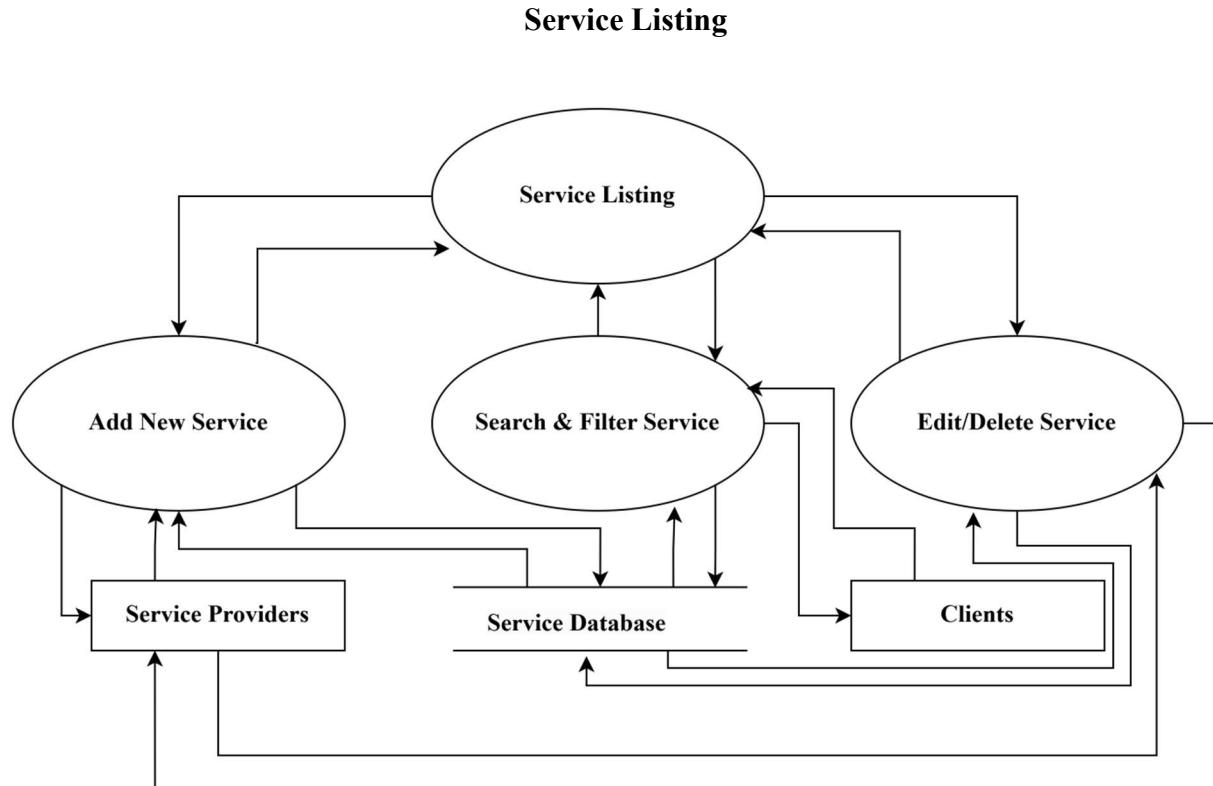
### Level-0



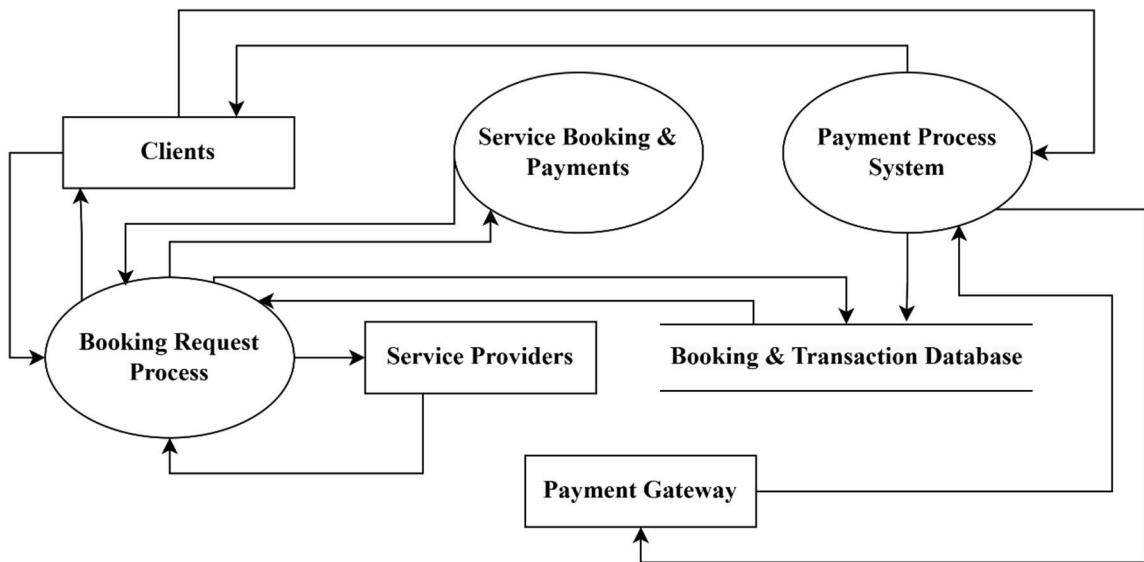
**Level-1**

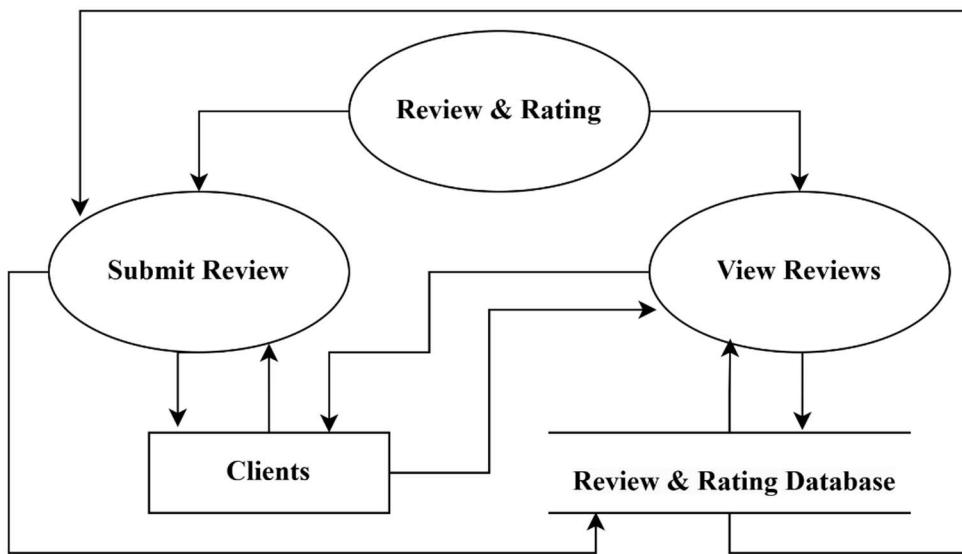
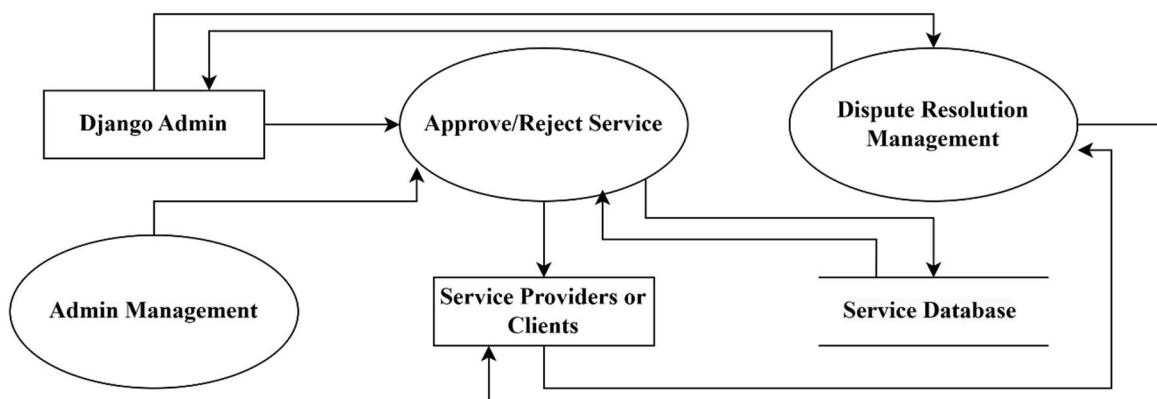
**Level-2****User Management**

## Level-2 Continues



## Service Booking &amp; Payments



**Level-2 Continues****Review & Rating****Admin Management**

## 6. DEVELOPMENT & TESTING

The development phase will follow an agile methodology, with iterative cycles of planning, development, and testing. Key activities will include:

- **Development:** Writing and integrating code based on the specified requirements and design.
- **Testing:** Conducting unit tests, integration tests, and user acceptance tests to ensure functionality, performance, and security.
- **Deployment:** Deploying the website to a production environment and ensuring it is fully operational.
- **Maintenance:** Providing ongoing support and updates based on user feedback and emerging requirements.

## 7. MILESTONES

<b>Sr. No</b>	<b>Project activity</b>	<b>Estimated Start Date</b>	<b>Estimated End Date</b>
1.	Project Allotment	08/01/2025	11/01/2025
2.	Requirement Gathering	11/01/2025	15/01/2025
3.	Synopsis Creation	15/01/2025	23/01/2025
4.	Synopsis Submission	24/01/2025	24/01/2025
5.	Designing	25/01/2025	05/02/2025
6.	Development	06/02/2025	28/02/2025
7.	Testing	01/03/2025	07/03/2025
8.	Finalizing the Project	08/03/2025	12/03/2025

## 8. MEETINGS WITH THE SUPERVISOR

Date of the meet	Comments by the supervisor	Signature of the supervisor
10/01/2025	Project topic discussions	
22/01/2025	Corrections in Synopsis	
23/01/2025	Checked the revised Synopsis and approved for presentation	

## 9. REFERENCES

- [1] GeeksforGeeks, "Levels in Data Flow Diagrams (DFD)," *GeeksforGeeks*, Available: <https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/>.
- [2] Diagrams.net, "Diagrams.net (Formerly Draw.io)," *Diagrams.net*, Available: <https://app.diagrams.net/>.

# **PROJECT REPORT**

## 1. INTRODUCTION

**EduMart** is a dynamic web-based platform designed to connect individuals seeking educational services with skilled professionals who offer them. The platform enables clients to discover, book, and procure a diverse range of services such as subject tutorials, graphic designing, application development, and more. At the same time, it empowers service providers to showcase their expertise, manage their service listings, communicate with clients, and handle bookings efficiently.

The project focuses on creating a user-friendly, secure, and scalable web application that facilitates smooth interactions between service providers and clients. **EduMart** incorporates modern features including real-time chat, booking and payment integration, and a review and rating system, ensuring a seamless service experience. The platform uses Django for the backend, HTML/CSS/JS for the frontend, and SQLite as the database.

By offering capabilities like advanced search filters, role-based user access, profile management, and automated booking status tracking, **EduMart** simplifies the process of both providing and accessing educational services. It serves as a reliable and centralized hub for educational needs in today's digital age.

**Project Title:** ‘EduMart’ (Marketplace for Educational Services)

**Category:** Web Development

## 1.1. PROBLEM DEFINITION

In the current scenario, individuals seeking educational or skill-based services face challenges in finding reliable and credible service providers. Traditional methods such as advertisements or word-of-mouth referrals prove to be time-consuming and often lack transparency regarding service quality. Likewise, skilled professionals struggle to effectively reach their target audience and showcase their expertise.

Most existing platforms fail to offer a specialized focus on educational services, leading to a cluttered user experience with irrelevant or non-academic offerings. Additionally, the absence of features like streamlined booking processes, secure online payments, and real-time communication creates inefficiencies and reduces trust between clients and service providers.

Clients often find it difficult to compare service providers based on important factors such as skill set, experience, pricing, and user feedback. On the other hand, service providers lack tools to highlight their unique strengths, manage bookings efficiently, and receive meaningful client reviews.

**EduMart** addresses these challenges by offering a dedicated, user-friendly, and transparent platform that caters exclusively to educational services. It simplifies the process of discovering, booking, and availing services, while empowering providers to showcase their offerings, manage client interactions, and build their reputation in a secure digital environment.

## 1.2. MOTIVATION

**EduMart** is driven by the need to bridge the gap between clients seeking educational or skill-based services and the professionals offering them. In the current digital landscape, individuals still struggle to find reliable, high-quality educational service providers. The process remains inefficient and scattered, especially on platforms that are not tailored for education-specific needs. These platforms often lack proper categorization, user-friendly interfaces, and the tools necessary for evaluating provider credibility.

Service providers, particularly educators, developers, and designers, also face challenges in showcasing their expertise, managing their bookings, and connecting with potential clients. Traditional methods do not offer features such as verified reviews, real-time interaction, or a central space to highlight specialized skills.

The motivation behind **EduMart** is to create a dedicated marketplace focused exclusively on educational and skill-based services. The platform aims to provide a structured, secure, and transparent environment where users can easily discover, compare, and book services, while professionals can manage their profiles, receive feedback, and build a strong reputation.

By incorporating features such as role-based user accounts (client/provider), categorized service listings, real-time chat, secure booking and payment systems, and a review/rating system, EduMart eliminates the inefficiencies of traditional methods and replaces them with a seamless, digital-first experience.

This platform empowers users with better decision-making tools and enables professionals to grow their reach in a competitive digital space. Ultimately, **EduMart** makes educational services more accessible, organized, and trustworthy for everyone involved.

## 2. OBJECTIVE

**EduMart** aims to create a secure, efficient, and user-friendly web platform that connects clients with skilled professionals offering educational and skill-based services. The platform ensures seamless accessibility for discovering, booking, and availing services, while empowering service providers to showcase their expertise, manage listings, and handle bookings effectively. With features such as advanced search filters, real-time booking availability, secure online payments, in-app chat, and a review and rating system, **EduMart** streamlines the entire service exchange process to enhance user experience, build trust, and foster transparency in every interaction. Prioritizing data security, scalability, and system reliability, **EduMart** safeguards user information while supporting growing traffic and service volume without compromising performance. Its ultimate objective is to deliver a dedicated, customized, and robust marketplace for clients and providers in the educational services space.

### 3. GOALS

The main goals of this project are:

1. **Centralized Marketplace:** Build a platform connecting clients with educational and skill-based service providers.
2. **Enhanced User Experience:** Create a user-friendly interface for seamless navigation and bookings.
3. **Trust and Transparency:** Implement reviews and ratings to ensure accountability and quality services.
4. **Empower Providers:** Enable professionals to showcase expertise and manage bookings efficiently.
5. **Streamlined Processes:** Simplify service discovery and bookings with advanced tools and real-time updates.
6. **Security:** Protect user data and payments with secure authentication and encryption.
7. **Customization:** Offer personalized profiles and tailored listings for better user relevance.
8. **Scalability:** Design the platform to handle growth without compromising performance.
9. **Real-Time Communication:** Integrate messaging for seamless client-provider interactions.
10. **Continuous Improvement:** Monitor feedback and update features to enhance functionality.

## 4. REQUIREMENT ANALYSIS & SPECIFICATIONS

### 4.1. Functional Requirements

- **User Registration and Authentication:** EduMart allows users to register as either clients or service providers. Secure login and registration are implemented using email and password authentication. Each user can manage their profile details.
- **Service Listings:** Service providers can create and manage listings by adding titles, descriptions, categories (like tutorials, design, or development), pricing, and availability. Listings are stored and updated dynamically in the database.
- **Advanced Search and Filters:** Clients can search for services using various filters such as category, pricing, availability, and user ratings to find the most relevant offerings.
- **Real-Time Booking:** Clients can book services based on the provider's availability. Once a booking is confirmed, both users receive notifications and the schedule is updated in real time.
- **Payment Gateway Integration:** Secure payment processing is integrated to handle transactions between clients and providers (implemented using Razorpay).
- **Review and Rating System:** After a service is completed, clients can submit reviews and rate the service provider. Ratings are visible to other users and contribute to the provider's public profile.
- **Messaging System:** A built-in real-time chat feature enables direct communication between clients and service providers to discuss service details or ask questions before confirming a booking.
- **Django Admin Panel:** The Django admin interface provides administrators with access to manage user accounts, monitor services, approve or remove listings, handle disputes, and maintain platform integrity.
- **User Profiles:** Each user has a dedicated profile. Clients can track booking history and upcoming services, while providers can manage listings, view reviews, and update portfolio content.

## 4.2. Non-Functional Requirements

- **Performance:** The platform is optimized for fast response times during browsing, searching, booking, and chatting, ensuring smooth operation even with multiple concurrent users.
- **Usability:** EduMart features a clean, intuitive interface with responsive design, making it accessible across desktops, tablets, and mobile devices.
- **Scalability:** The architecture is designed to support future expansion, enabling more users, services, and features to be added without degrading performance.
- **Security:** EduMart ensures user data is securely stored. Passwords are encrypted, sessions are managed through middleware, and unauthorized access is restricted. Payment data is handled through secure protocols (when integrated).
- **Compatibility:** The platform is tested and works across major web browsers like Chrome, Firefox, Brave, and Safari, and is compatible with Windows, macOS, and Linux.
- **Maintainability:** Clean, modular Django code and comprehensive inline documentation enable developers to easily maintain and extend the platform.
- **Backup and Data Recovery:** Regular database backups are recommended for production deployment to prevent data loss in case of system failure.
- **Accessibility:** The platform follows basic accessibility guidelines, with support for keyboard navigation and screen readers where applicable.

### 4.3. Technical Requirements

#### 1. Frontend Technologies

- **HTML:** Creating Structure and markup for webpages.
- **CSS:** Styling and layout using Custom CSS.
- **JavaScript:** Client-side scripting for interactive features and game logic.
- **Bootstrap:** Framework for more prebuilt features for defining interactive UI.

#### 2. Backend Technologies

- **Django:** Framework for building the backend, handling routing, and interacting with the database.
- **Database:** SQLite for storing user data, game data, and performance metrics.
- **ORM:** Django's ORM (Object Relational Mapping) for database interactions.

## 4.4. SOFTWARE & HARDWARE REQUIREMENTS

### **Hardware Required:**

<b><u>Name of Components</u></b>	<b><u>Specifications</u></b>
Processor	: Intel/AMD/Ryzen
RAM	: 4GB or More
Speed	: 1.00 GHz
System Type	: 64-bit operating system

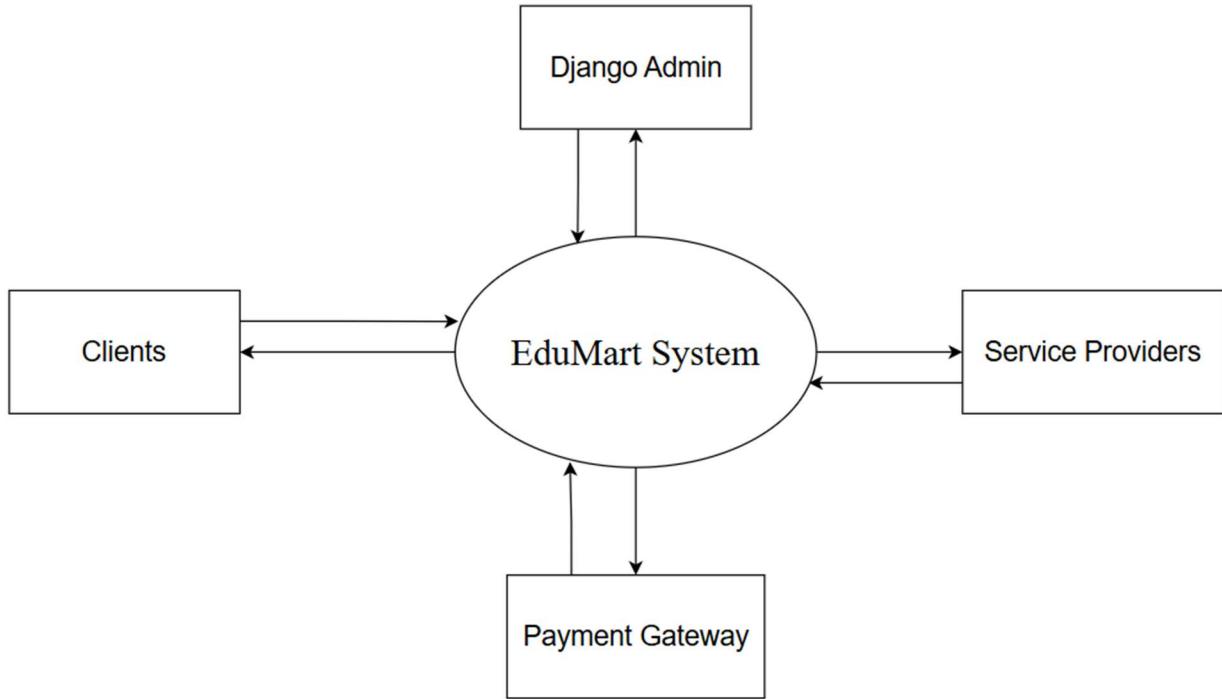
### **Software Required:**

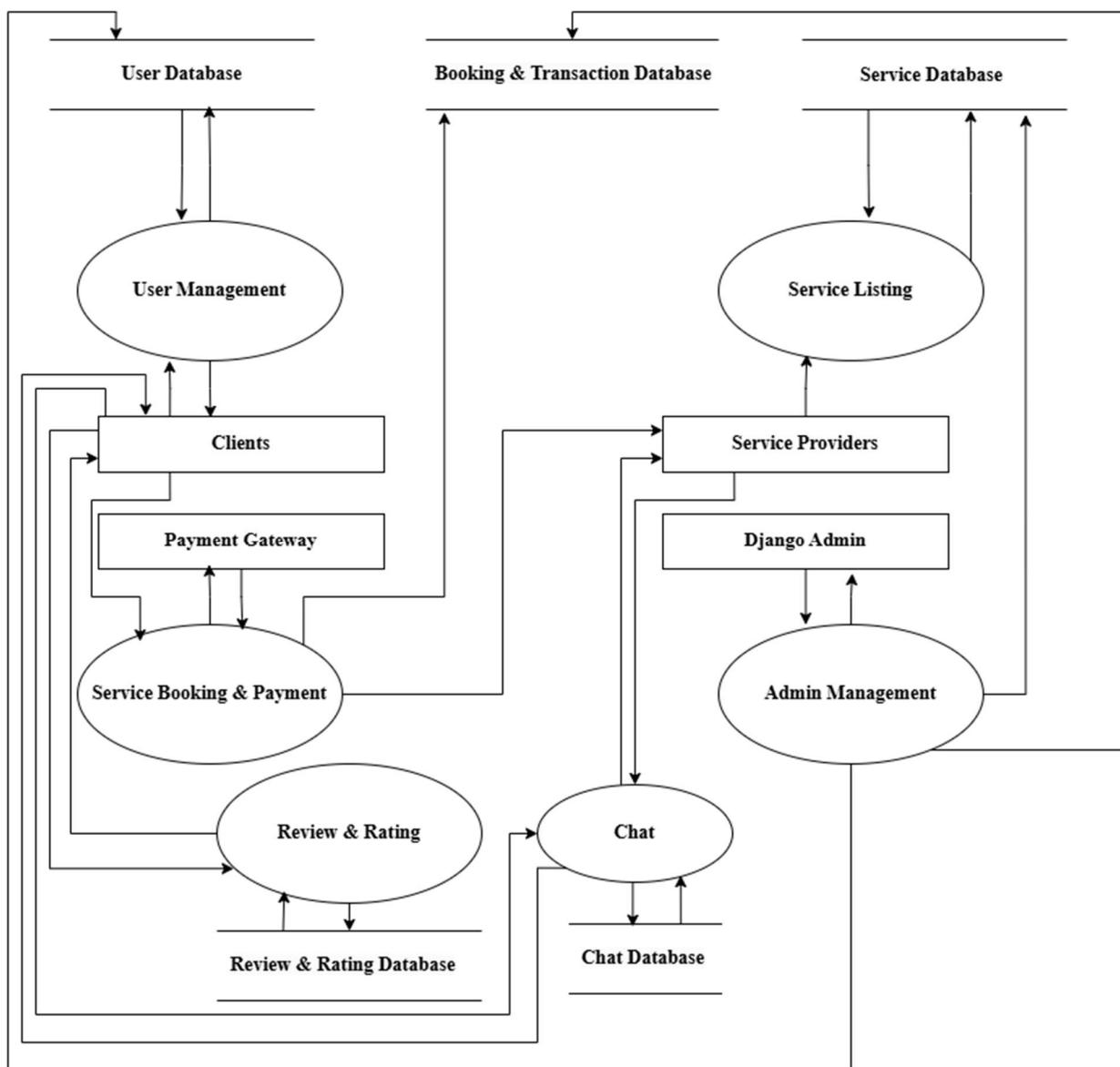
<b><u>Name of Components</u></b>	<b><u>Specifications</u></b>
Operating System	: Windows/Mac/Linux
Frontend	: HTML, CSS, JavaScript
Framework	: Django, Bootstrap
Database	: SQLite
IDE	: Visual Studio Code
Browser	: Chrome/Mozilla/Edge/Brave

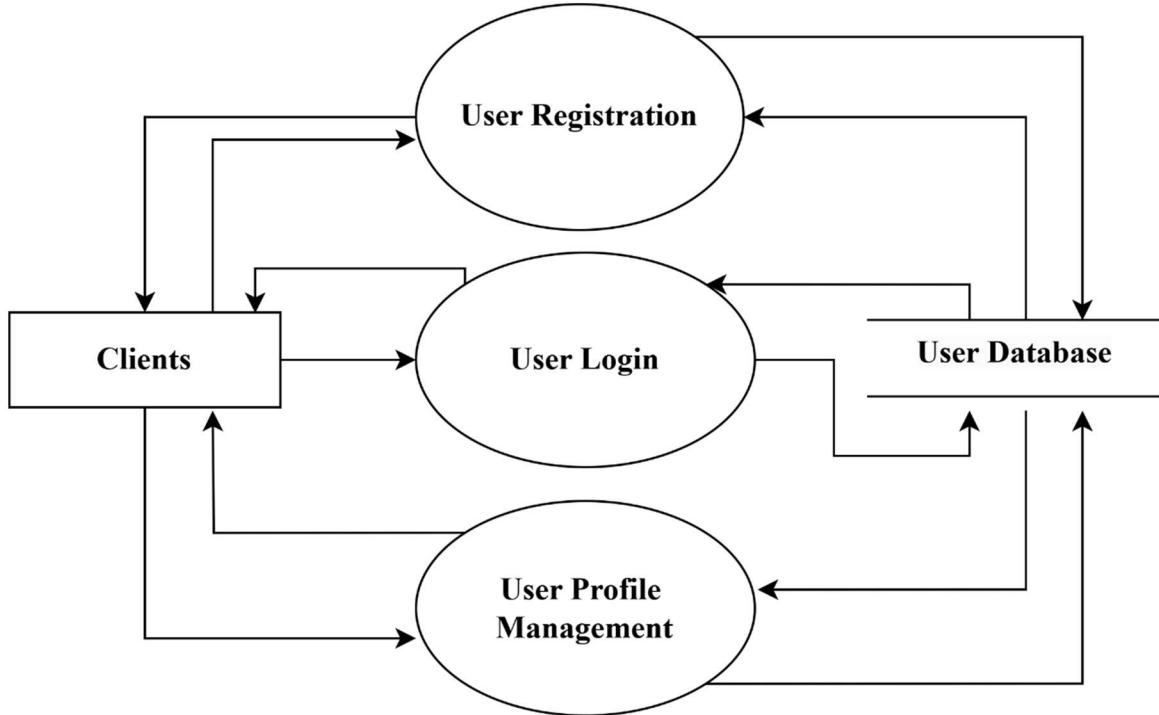
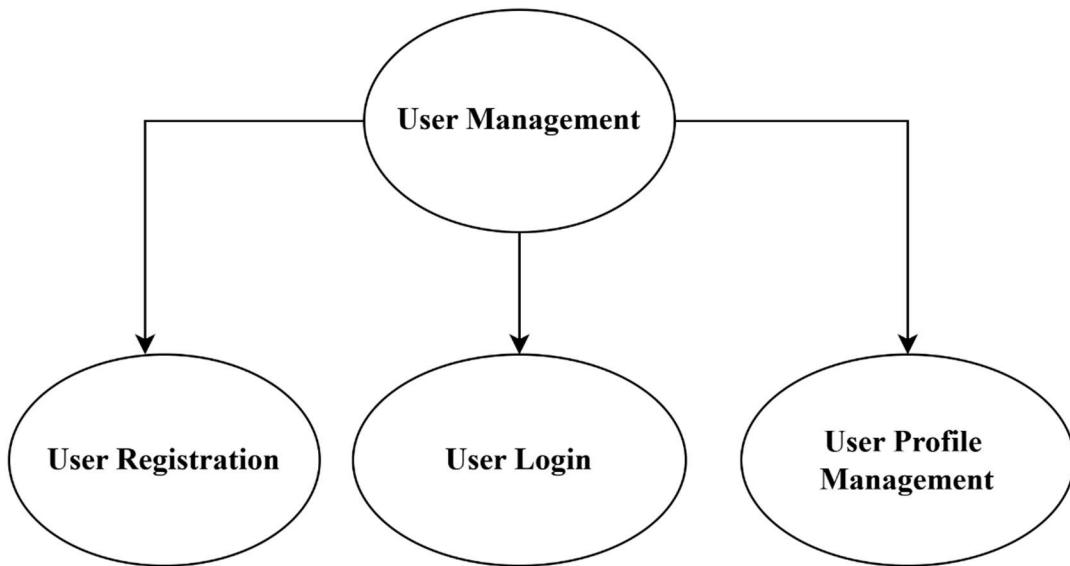
## 5. SYSTEM DESIGN

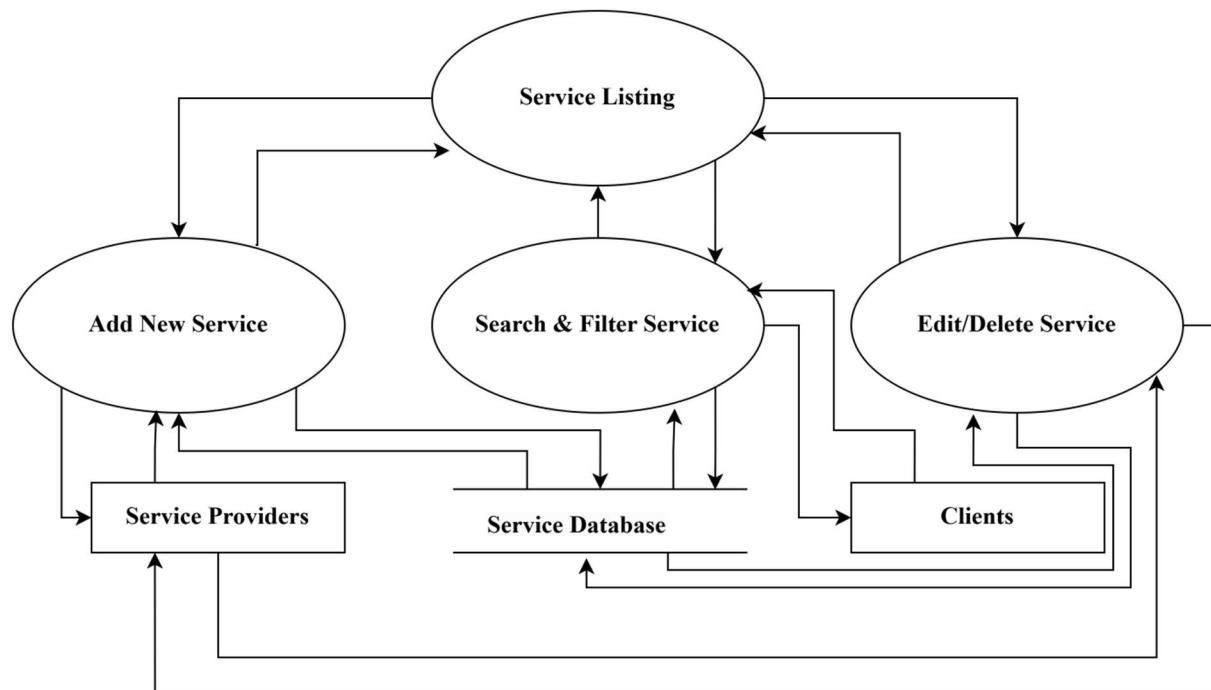
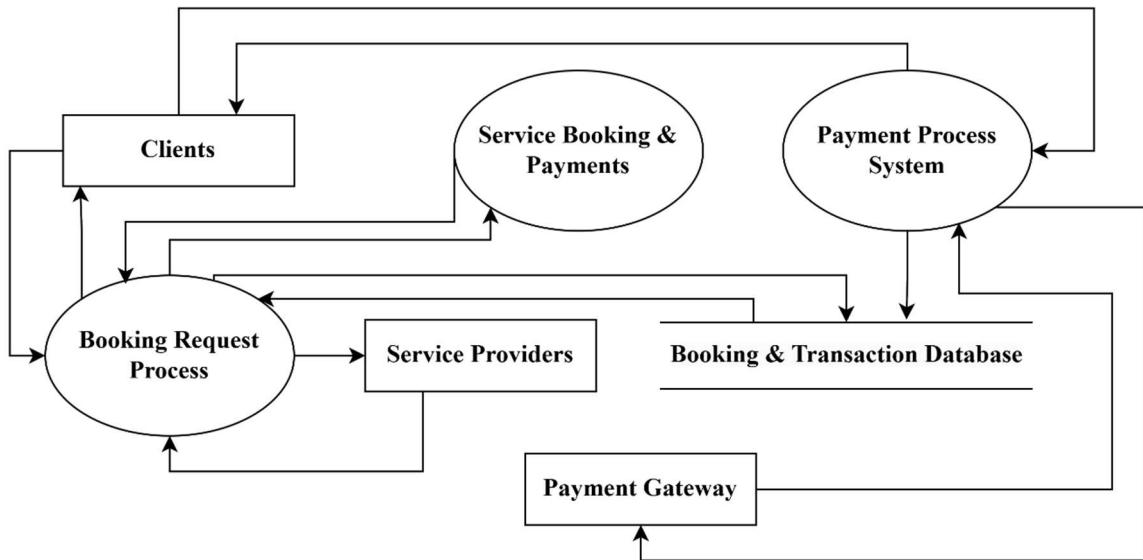
Detailed System design of **EduMart** using Data Flow Diagram (DFD):

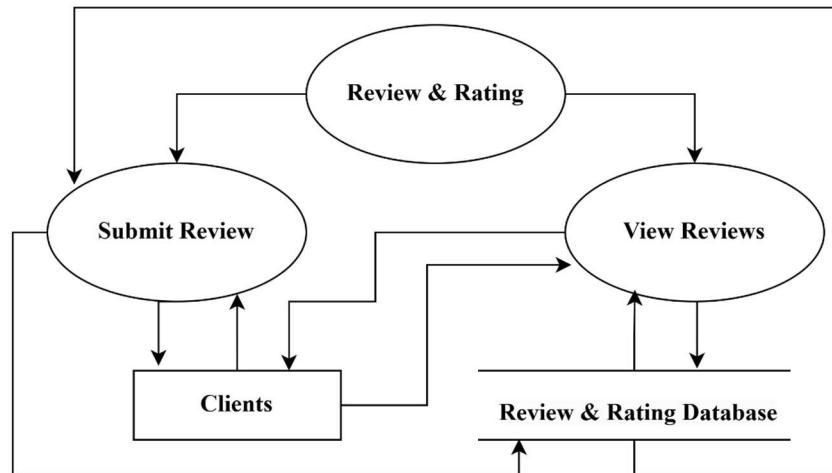
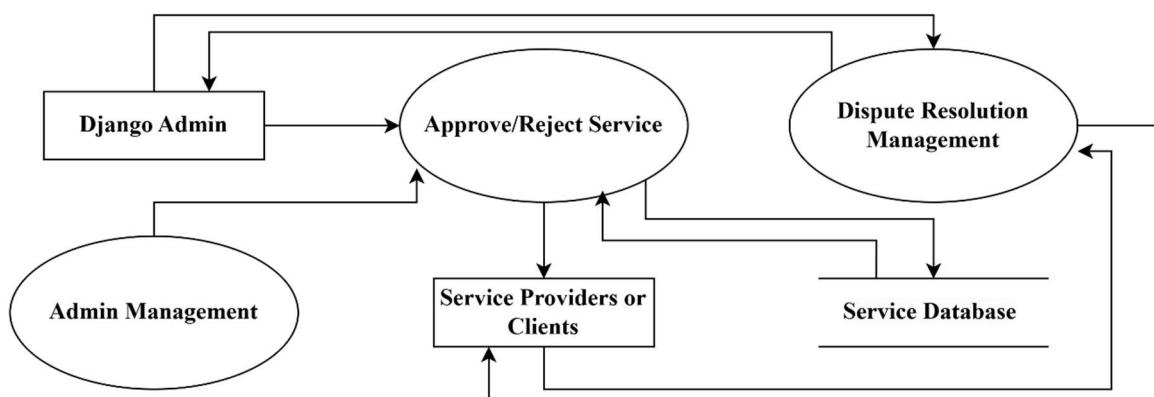
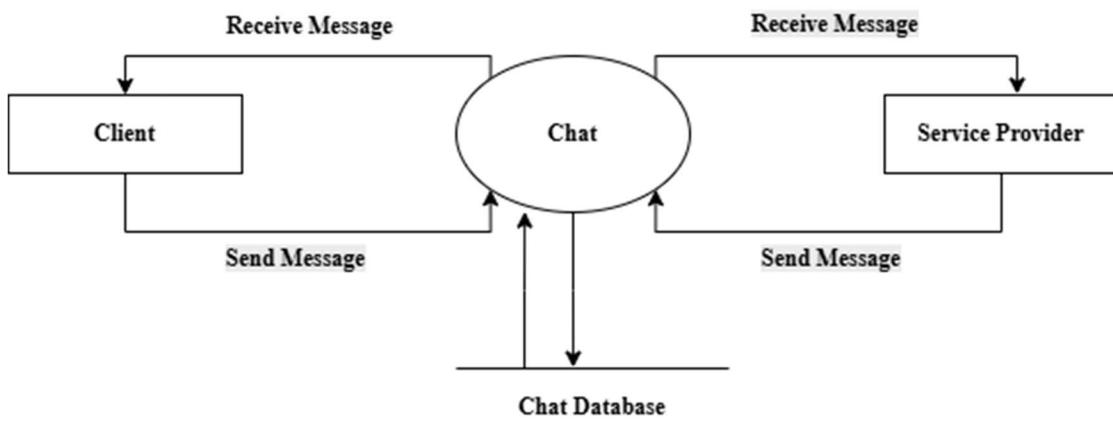
### Level-0



**Level-1**

**Level-2****User Management**

**Level-2 Continues****Service Listing****Service Booking & Payments**

**Level-2 Continues****Review & Rating****Admin Management****Chat**

## 6. DEVELOPMENT & TESTING

The development phase will follow an agile methodology, with iterative cycles of planning, development, and testing. Key activities will include:

- **Development:** Writing and integrating code based on the specified requirements and design.
- **Testing:** Conducting unit tests, integration tests, and user acceptance tests to ensure functionality, performance, and security.
- **Deployment:** Deploying the website to a production environment and ensuring it is fully operational.
- **Maintenance:** Providing ongoing support and updates based on user feedback and emerging requirements.

## 7. MODULES

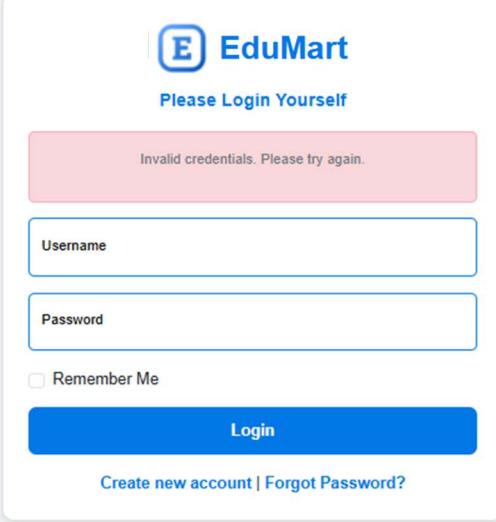
The **EduMart** project is structured into the following core modules:

- 1. User Authentication Module:** This module manages secure login and registration for all users, including clients and service providers. Role-based access control ensures that users only access features relevant to their role. Passwords are securely encrypted, and session management ensures data privacy.
- 2. Service Listing & Management Module:** Service providers can create, update, and manage educational service listings. Each service includes a title, detailed description, pricing, availability, and category (e.g., tutoring, design, development). Listings can be approved or moderated by the admin via the Django panel.
- 3. Search & Filter Module:** Clients can search for educational services using keywords and apply filters such as category, title and service provider name. This helps users quickly discover relevant services based on their needs.
- 4. Booking & Scheduling Module:** Clients can book services in real-time based on the provider's availability. This module handles booking confirmations, status tracking, and automatic updates to service schedules, ensuring a seamless user experience.
- 5. Payment Gateway Integration Module:** A secure payment gateway handles all financial transactions between clients and service providers. This module ensures payment confirmation, transaction tracking, and security compliance to protect user data.
- 6. Chat & Communication Module:** Clients and service providers can communicate directly through a real-time messaging system integrated within the platform.
- 7. Review & Rating Module:** After service completion, clients can submit reviews and ratings for the services they used. These ratings are publicly visible and help build trust and transparency within the platform.
- 8. Admin Management Module:** Admins can manage users, review service listings, monitor bookings, resolve disputes, and moderate platform activity using the Django admin interface. This module ensures the integrity and smooth operation of the entire platform.
- 9. User Profile & Dashboard Module:** Each user has access to a personalized dashboard that displays their bookings, listed services, reviews, and profile information. Service providers can manage their offerings, view client feedback, and update portfolios.

## 8. TEST CASES

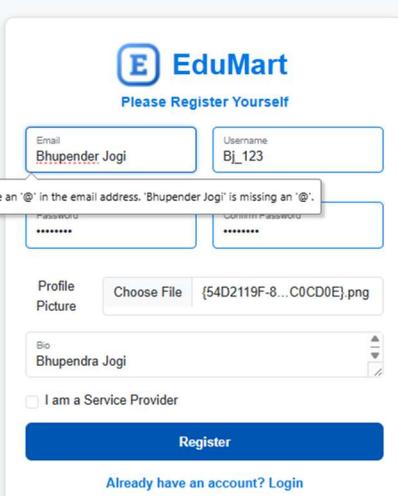
### 1. User Authentication

**Test Case 1.1:** User Login with Invalid Credentials



The screenshot shows the EduMart login interface. At the top is the EduMart logo and the text "Please Login Yourself". Below is a red error message box containing the text "Invalid credentials. Please try again.". There are two input fields: "Username" and "Password". Underneath them is a "Remember Me" checkbox and a blue "Login" button. At the bottom of the form are links for "Create new account" and "Forgot Password?".

**Test Case 1.2:** User Registration



The screenshot shows the EduMart registration interface. At the top is the EduMart logo and the text "Please Register Yourself". The form includes fields for "Email" (containing "Bhupender Jogi") and "Username" (containing "Bj\_123"). An orange error message box below the email field says "Please include an '@' in the email address. 'Bhupender Jogi' is missing an '@'." There are two password fields, "Password" and "Confirm Password", both containing "\*\*\*\*\*". Below these is a "Profile Picture" field with a chosen file named "54D2119F-8...C0CD0E.png". A "Bio" field contains "Bhupendra Jogi". A checkbox for "I am a Service Provider" is unchecked. At the bottom is a blue "Register" button and a link for "Already have an account? Login".

EduMart  
Please Register Yourself

Passwords do not match.

Email: bhupendrajogi420@gmail.com  
Username: BJ\_123

Password:  Confirm Password:

Profile Picture: Choose File Screenshot 2... 15 110116.jpg

Bio: Bhupendra Jogi

I am a Service Provider

**Register**

Already have an account? [Login](#)

EduMart  
Please Register Yourself

A user with this email already exists.

Email: deepaksingh0507097@gmail.com  
Username: deepak\_97

Password:  Confirm Password:

Profile Picture: Choose File Screenshot 2... 15 110116.jpg

Bio: I am here to have some services

I am a Service Provider

**Register**

Already have an account? [Login](#)

### Test Case 1.3: Invalid OTP (Forgot Password/Registration Verification)

EduMart  
Verify Your OTP

We have sent a verification code to your email. Please enter it below.

Invalid OTP.

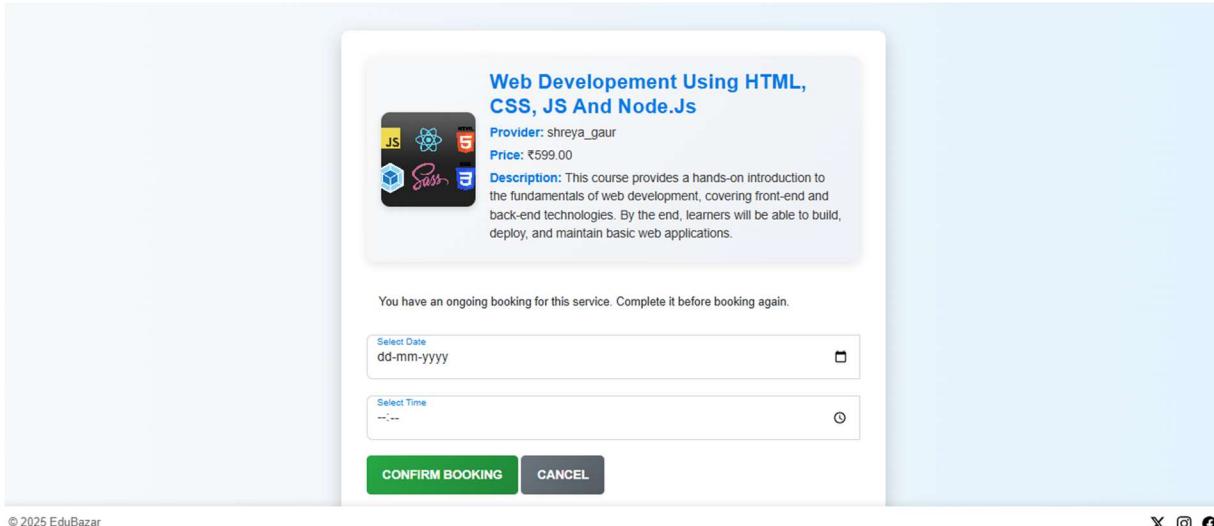
Enter OTP: 564646

**Verify OTP**

[Resend OTP](#)

## 2. Booking System

### Test Case 2.1: For Ongoing Service



© 2025 EduBazar

X

## 9. PROJECT OUTPUTS

### Home Page

**Welcome to EduMart - Your Marketplace for Educational Services**

Find expert tutors, designers, developers, and more – all in one place!

**Why Choose EduMart?**

- Trusted Professionals**  
Verified tutors & experts.
- Easy Booking**  
Schedule sessions at your convenience.
- Secure Payments**  
Pay safely with integrated gateways.
- Real Reviews**  
Ratings & feedback from real users.

**Popular Services**

**Tutoring**  
Math, Science, Coding, and more.

**Graphic Design**  
Logos, UI/UX Illustrations.

**App Development**  
Websites, mobile apps, software.

**How It Works?**

A simple 4-step process to connect with the best educational services.

**Sign Up**  
Create your free account

**Explore Services**  
Find what you need

**Book & Pay**  
Secure your booking

**Learn & Grow**  
Get expert help!

## Registration

EduMart  
Please Register Yourself

Email: shreyagaur9454@gmail.com    Username: shreya\_gaur

Password: .....    Confirm Password: .....

Profile Picture: Choose File: Shreya g.jpg

Bio: Web Developer

I am a Service Provider

**Register**

Already have an account? [Login](#)

EduBazar  
Verify Your OTP

We have sent a verification code to your email. Please enter it below.

OTP sent to your email. Please verify.

Enter OTP: 042779

**Verify OTP**  
[Resend OTP](#)

## Login

EduMart  
Please Login Yourself

Username: shreya\_gaur

Password: .....

Remember Me

**Login**

[Create new account](#) | [Forgot Password?](#)

## User Profile (Login)

The screenshot shows the user profile page for 'shreya\_gaur'. At the top, there's a search bar and navigation links for 'Browse Services', 'Add New Service', and 'Logout'. The profile section features a circular profile picture, the provider's name 'shreya\_gaur', their email 'shreyagaur9454@gmail.com', and their bio 'Web Developer'. A yellow 'Edit Profile' button is present. Below this is a 'Booking Requests' section stating 'No new booking requests.' Underneath is a 'Services You Have Booked' section, which displays a banner for 'UX/UI WEB DESIGN' against a blue background with circuit board patterns. The footer includes a copyright notice '© 2025 EduMart' and social media icons for X, Instagram, and Facebook.

## Edit Profile

The 'Edit Your Profile' form is displayed. It includes fields for 'Email' (shreyagaur9454@gmail.com), 'Username' (shreya\_gaur), and 'Bio' (Web Developer). There's also a 'Profile Picture' section with a file input field ('Choose File') showing 'No file chosen'. A large blue 'Save Changes' button is at the bottom, along with a link to 'Back to Profile'.

## Add Service

EduMart

Search by title, category, or provider.

Create Service

Service Title  
Video Editing

Description  
"Crafting engaging visuals by trimming, enhancing, and assembling video clips into a polished final product."

Category  
Tutoring

Price (INR ₹)  
499

Service Image  
Choose File timeline-video-video-editing-video-edits.webp

Available

Create Service Cancel

© 2025 EduMart



## Added Service

EduMart

Search by title, category, or provider.

Your Services

**Front End Development**  
Category: Tutoring  
Price: ₹299.00  
Availability:  Available  
Created on: Mar 17, 2025  
Description: Front End Development Using HTML, CSS, JavaScript, React, Node, Express etc.  
Edit Delete View Reviews

**Django Tutorial**  
Category: Tutoring  
Price: ₹3000.00  
Availability:  Available  
Created on: Mar 18, 2025  
Description: Django Tutorial For Beginners  
Edit Delete View Reviews

**Video Editing**  
Category: Tutoring  
Price: ₹499.00  
Availability:  Available  
Created on: Apr 07, 2025  
Description: "Crafting engaging visuals by trimming, enhancing, and assembling video clips into a polished final product."  
Edit Delete View Reviews

Reviews for Your Services

Service: Front End Development  
Provider: dinesh\_239

© 2025 EduMart



## Service Listing

EduMart Search by title, category, or provider... Search Browse Services Add New Service dinesh\_239 Logout

### Available Services



**Front End Development**

Category: Tutoring  
Description: Front End Development Using HTML, CSS, JavaScript, React, Node, Express etc.

**Price:** ₹299.00  
**Provider:** dinesh\_239

**Available** View Provider Profile



**Django Tutorial**

Category: Tutoring  
Description: Django Tutorial For Beginners

**Price:** ₹3000.00  
**Provider:** dinesh\_239

**Available** View Provider Profile



**React Framework**

Category: Web Development  
Description: React is a JavaScript library (not a full framework) developed by Facebook (Meta) for building fast and interactive user interfaces, ...

**Price:** ₹499.00  
**Provider:** sweta\_1509

**Available** View Provider Profile



**Cloud Computing**

Category: Tutoring  
Description: Cloud computing is a scalable services consumption and delivery platform that provides on-demand computing service for shared pool of resources, ...

**Price:** ₹1499.00  
**Provider:** abhi\_no

**Available** View Provider Profile



**Web Development Using HTML, CSS, JS and Node.js**

Category: Web Development  
Description: This course provides a hands-on introduction to the fundamentals of web development, covering front-end and back-end technologies. By the end, ...

**Price:** ₹599.00  
**Provider:** shreya\_gaur

**Available** View Provider Profile



**Node Js**

Category: Web Development  
Description: Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to run JavaScript code on the server side. It ...

**Price:** ₹799.00  
**Provider:** sweta\_1509

**Available** View Provider Profile

© 2025 EduMart X @ F

## Searching

EduMart video Search Browse Services Add New Service dinesh\_239 Logout

### Search Results for "video"



**Video Editing**

Category: Tutoring  
Description: "Crafting engaging visuals by trimming, enhancing, and assembling video clips into a polished final product."

**Price:** ₹499.00  
**Provider:** dinesh\_239

**Available** View Provider Profile

© 2025 EduMart X @ F

## Service Booking (By Client)

The screenshot shows a service listing for "Video Editing" provided by "dinesh\_239" at a price of ₹499.00. The description states: "Crafting engaging visuals by trimming, enhancing, and assembling video clips into a polished final product." Below the service details, there are input fields for "Select Date" (09-04-2025) and "Select Time" (12:40). A message box indicates "0 clients have already booked this slot." At the bottom are "CONFIRM BOOKING" and "CANCEL" buttons.

## Accepting Booking Request (By Service Provider)

The screenshot shows a booking request from "sagar\_139" for "Video Editing" on April 9, 2025, at 12:40 p.m. The status is "Pending". The "Accept" button is highlighted in green, while the "Reject" button is red. Below the request, there is a section titled "Services You Have Booked".

## Payment (By Client)

The screenshot shows a booking history entry for "Video Editing" by "dinesh\_239" for "sagar\_139" on April 7, 2025, at 16:44 PM. The status is "Approved". At the bottom, there are "Make Payment" and "Cancel" buttons.

Secure Payment

Test Mode Instructions  
Use these test card details:

<b>Success Card</b>	<b>Failure Card</b>
Card Number: 4111 1111 1111 1111	Card Number: 4242 4242 4242 4242
CVV: 123	CVV: 123
Expiry: 12/25	Expiry: 12/25

**Booking Details**

Service: Video Editing
Provider: dinesh_239
Amount: ₹499.00

**Pay Securely**    **Cancel**

© 2025 EduMart

Payment Options

- Recommended
- UPI
- Cards
- Netbanking
- Wallet
- Pay Later

Add a new card

4111 1111 1111 1111 VISA

12 / 25

Sagar Vishwa

Save this card as per RBI guidelines

**Continue**

© 2025 EduMart

Payment Options

Enter OTP to complete payment

Enter OTP sent to the number linked to your card ending with 1111

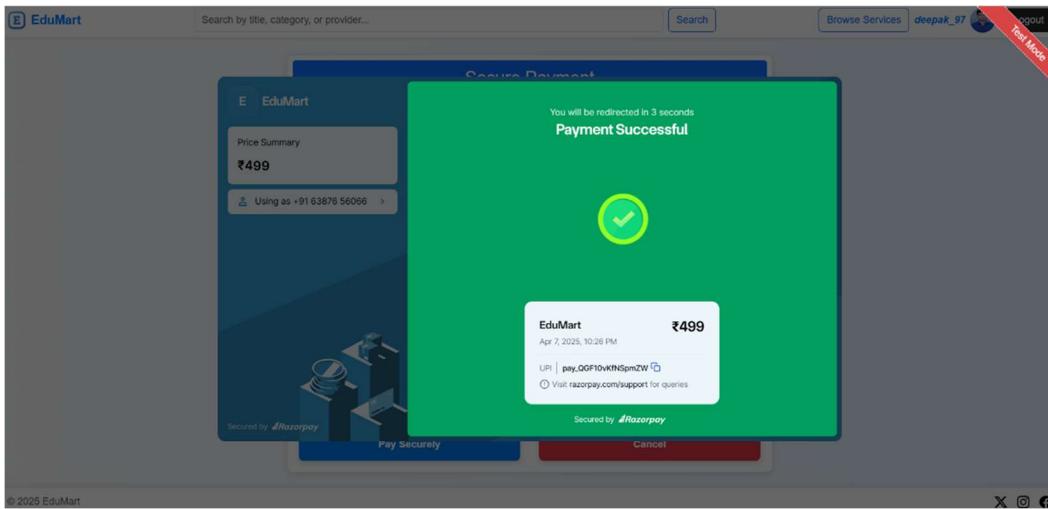
053269

Pay on bank's page Resend OTP In 21s

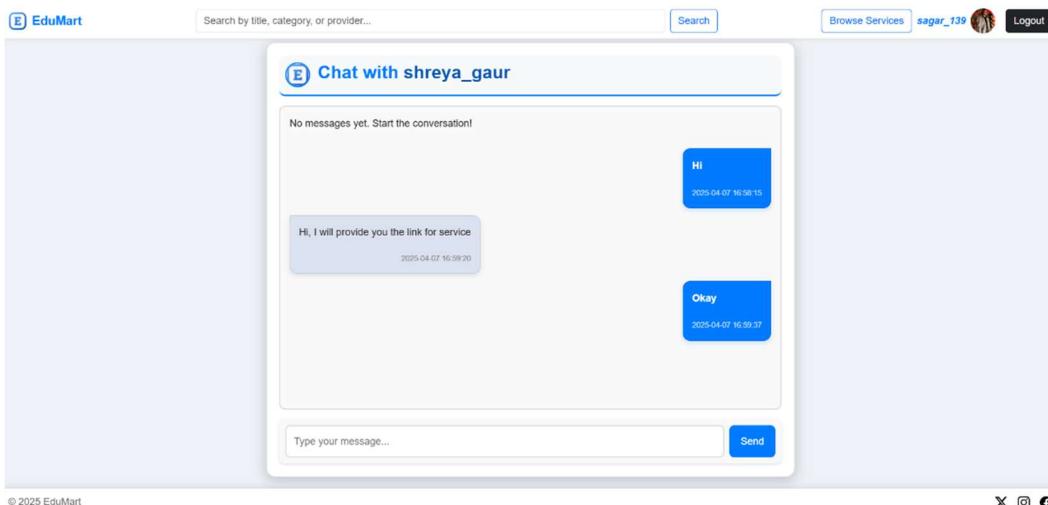
**Continue**

© 2025 EduMart

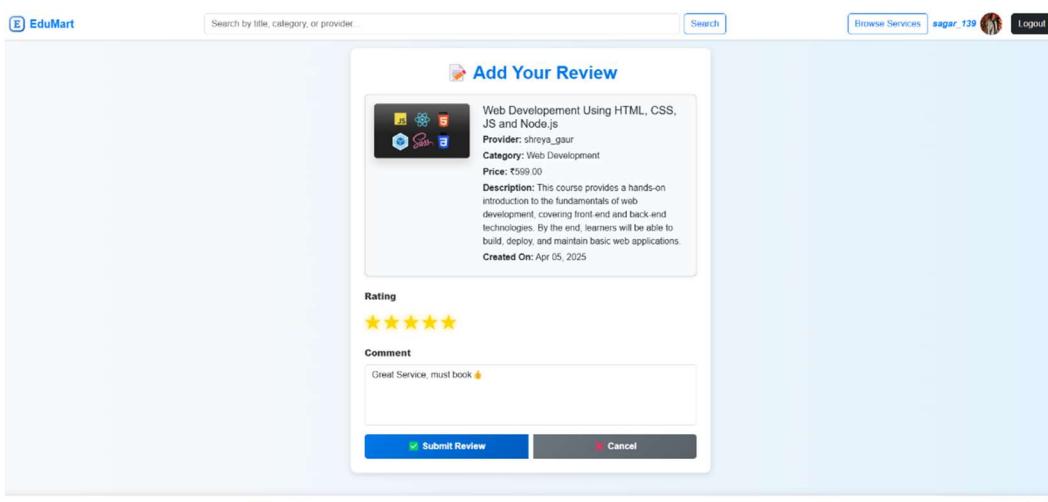
## Payment Success



## Client, Service Provider Interaction (Chat)



## Submitting Review (By Client when Service Completes)



## Review

The screenshot shows a user's review history. The first review is for a 'React Framework' service provided by 'sweta\_1509'. The reviewer is 'sagar\_139' with a rating of 5/5. The comment is: 'Excellent React course! Clear explanations, hands-on projects, and strong coverage of concepts like components, hooks, and state management. Perfect for beginners and developers looking to build real-world apps.' It was reviewed on April 05, 2025, at 09:48 AM. The second review is for a 'Web Development Using HTML, CSS, JS and Node.js' service provided by 'shreya\_gaur'. The reviewer is 'sagar\_139' with a rating of 5/5. The comment is: 'Great Service, must book.' It was reviewed on April 07, 2025, at 05:02 PM.

## Viewing Provider Profile

The screenshot shows a provider's profile for 'sweta\_1509'. The profile includes a circular profile picture of a person, the provider's name 'sweta\_1509', their email 'swetapandey150902@gmail.com', and a bio: 'I am Master Student and future scholar'. There is a 'Edit Profile' button. Below the profile is a section for 'Booking Requests' which states 'No new booking requests'. Under 'Your Services', there are two services listed: 'React Framework' and 'Node Js'. The 'React Framework' service has a blue background with a React logo, a price of ₹499.00, and availability status. The 'Node Js' service has a green background with a Node.js logo, a price of ₹799.00, and availability status. Both services have 'Edit', 'Delete', and 'View Reviews' buttons. At the bottom, there is a section for 'Reviews for Your Services' showing the same review as the previous screenshot for the 'React Framework' service.

## View Review

The screenshot shows a web page titled "REVIEWS FOR 'REACT FRAMEWORK'". At the top, there is a logo for "React" featuring a stylized atom or gear icon. Below the logo, the title "React Framework" is displayed in bold. Underneath the title, it says "Provider: sweta\_1509", "Category: Web Development", and "Price: ₹499.00". A detailed description follows: "React is a JavaScript library (not a full framework) developed by Facebook (Meta) for building fast and interactive user interfaces, especially single-page applications (SPAs)." The date "Created On: Mar 28, 2025" is also mentioned. Below this section, there is a heading "User Reviews" with a small icon. It lists a review from a client named "sagar\_139" dated "Apr 05, 2025 at 09:48 AM". The rating is 5/5 stars. The comment reads: "Excellent React course! Clear explanations, hands-on projects, and strong coverage of concepts like components, hooks, and state management. Perfect for beginners and developers looking to build real-world apps." At the bottom of the review section is a blue button labeled "Back to Profile". The footer of the page includes the copyright notice "© 2025 EduMart" and social media icons for X, Instagram, and Facebook.

## 10. CODES

### edumart (Project App)

#### **asgi.py**

```
import os
from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack
from chat.routing import websocket_urlpatterns
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "EduMart.settings")
application = ProtocolTypeRouter({
    "http": get_asgi_application(),
    "websocket": AuthMiddlewareStack(
        URLRouter(websocket_urlpatterns)
    ),
})
```

#### **settings.py**

```
import os
import random
import string
from pathlib import Path
BASE_DIR = Path(__file__).resolve().parent.parent
SECRET_KEY = 'django-insecure-zer*d3zpq!vg^2n(i43co3rulxmb!=ldtuw)w+_fvbe&d34+h'
DEBUG = True
ALLOWED_HOSTS = []
AUTH_USER_MODEL = 'users.CustomUser'
LOGIN_URL = '/users/login/'
RAZORPAY_API_KEY = "rzp_test_A1QJ17pPhtyAce"
RAZORPAY_API_SECRET = "L5i7LWxeEmsISKFq1e5xFmiQ"
SESSION_SAVE_EVERY_REQUEST = True
SESSION_COOKIE_AGE = 86400 * 7
SESSION_COOKIE_SECURE = False
SESSION_ENGINE = "django.contrib.sessions.backends.db"
SESSION_COOKIE_NAME = "edumart_session"
SESSION_COOKIE_HTTPONLY = True
SESSION_COOKIE_SECRET = ".join(random.choices(string.ascii_letters + string.digits,
k=32))"
INSTALLED_APPS = [
    'daphne',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
```

```

'channels',
'users',
'services',
'bookings',
'payments',
'reviews',
'chat',
]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'EduMart.middleware.auth_middleware.AuthMiddleware',
    'EduMart.middleware.session_middleware.UniqueSessionMiddleware',
]
ROOT_URLCONF = 'edumart.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / "templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
ASGI_APPLICATION = "edumart.asgi.application"
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels.layers.InMemoryChannelLayer",
    },
}
WSGI_APPLICATION = edumart.wsgi.application
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

```

```

        }
    }
AUTH_PASSWORD_VALIDATORS = [
    {'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator'},
    {'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator'},
    {'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator'},
    {'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator'},
]
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True
STATIC_URL = 'static/'
STATICFILES_DIRS = [BASE_DIR / "static"]
MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / "media"
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'davidsilvermann309@gmail.com'
EMAIL_HOST_PASSWORD = 'jemuebwmpganqgiv'
DEFAULT_FROM_EMAIL = 'davidsilvermann309@gmail.com'

urls.py
from django.contrib import admin
from django.urls import path,include
from django.conf import settings
from django.conf.urls.static import static
from .views import index, profile, provider_profile, manage_booking
urlpatterns = [
    path('admin/', admin.site.urls),
    path("",index,name="index"),
    path('profile',profile,name="profile"),
    path('provider_profile/<int:provider_id>', provider_profile, name="provider_profile"),
    path('booking/<int:booking_id>/<str:action>', manage_booking, name='manage_booking'),
    path('users/',include('users.urls')),
    path('services/', include('services.urls')),
    path('bookings/',include('bookings.urls')),
    path('payments/', include('payments.urls')),
    path('reviews/', include('reviews.urls')),
    path('chat/', include('chat.urls')),
]
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

views.py

```

```

from django.shortcuts import render, get_object_or_404, redirect
from django.contrib.auth.decorators import login_required
from users.models import CustomUser
from services.models import Service
from reviews.models import Review
from bookings.models import Booking # Import the Booking model
from django.contrib import messages
from django.utils import timezone
def index(request):
    return render(request, 'index.html')
@login_required
def profile(request):
    delete_past_bookings() # Delete outdated bookings
    user = request.user
    services = Service.objects.filter(provider=user, available=True) if user.is_service_provider
    else None
    client_bookings = Booking.objects.filter(client=user, status="Confirmed") if not
    user.is_service_provider else None
    bookings_client = Booking.objects.filter(client=user, status__in=["Pending", "Approved"]) if
    not user.is_service_provider else None
    provider_bookings = Booking.objects.filter(service_provider=user, status="Confirmed") if
    user.is_service_provider else None
    client_completed_services = Booking.objects.filter(client=user, status="Completed") if not
    user.is_service_provider else None
    provider_completed_services = Booking.objects.filter(service_provider=user,
    status="Completed") if user.is_service_provider else None
    notifications = Booking.objects.filter(service_provider=user, status="Pending") if
    user.is_service_provider else None
    client_reviews = Review.objects.filter(client=user) if not user.is_service_provider else None
    provider_reviews = Review.objects.filter(service_provider=user) if user.is_service_provider
    else None
    return render(request, "profile.html", {
        "user": user,
        "services": services,
        "client_bookings": client_bookings, # Confirmed bookings
        "bookings_client": bookings_client, # Pending & Approved bookings
        "provider_bookings": provider_bookings,
        "client_completed_services": client_completed_services,
        "provider_completed_services": provider_completed_services,
        "notifications": notifications,
        "client_reviews": client_reviews,
        "provider_reviews": provider_reviews,
    })
@login_required
def manage_booking(request, booking_id, action):
    booking = get_object_or_404(Booking, id=booking_id, service_provider=request.user)

```

```

if action == 'accept':
    booking.status = 'Confirmed'
    messages.success(request, f"Booking #{booking.id} has been confirmed!")
elif action == 'reject':
    booking.status = 'Cancelled'
    messages.warning(request, f"Booking #{booking.id} has been rejected!")
    booking.save()
return redirect('profile')
def provider_profile(request, provider_id):
    provider = get_object_or_404(CustomUser, id=provider_id)
    if request.user.is_authenticated and request.user == provider:
        return redirect('profile')
    services = Service.objects.filter(provider=provider, available=True)
    provider_reviews = Review.objects.filter(service_provider=provider).select_related('client', 'service')
    return render(request, 'provider_profile.html', {
        'provider': provider,
        'services': services,
        'provider_reviews': provider_reviews
    })
def delete_past_bookings():
    now = timezone.now()
    today = now.date()
    current_time = now.time()
    Booking.objects.filter(date__lt=today).exclude(status__in=["Confirmed", "Completed"]).delete()
    Booking.objects.filter(date=today, time__lt=current_time).exclude(status__in=["Confirmed", "Completed"]).delete()
    Booking.objects.filter(
        date__lt=today, status="Confirmed"
    ).update(status="Completed")
    Booking.objects.filter(
        date=today, time__lt=current_time, status="Confirmed"
    ).update(status="Completed")

```

**wsgi.py**

```

import os
from django.core.wsgi import get_wsgi_application
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'EduMart.settings')
application = get_wsgi_application()

```

**Static****css/index\_style.css**

```

body { font-family: 'Arial', sans-serif; background-color: #f8f9fa; color: #333; margin: 0; }
.navbar { background-color: #fff; padding: 10px 0; box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1); transition: background-color 0.3s ease, box-shadow 0.3s ease; }
.navbar.fixed-top { position: fixed; top: 0; left: 0; right: 0; z-index: 1000; }
.navbar.navbar-scroll { background-color: rgba(255, 255, 255, 0.9); backdrop-filter: blur(10px); }

```

```

.navbar-brand { font-weight: bold; font-size: 1.5rem; color: #007bff !important; }
.navbar-brand img { margin-right: 10px; }
.navbar p { font-size: 1.1rem; color: #333; margin-bottom: 0; white-space: nowrap; font-style: italic; }
.carousel { position: relative; margin-bottom: 50px; padding-top: 70px; }
.carousel-img { width: 100%; height: 80vh; object-fit: cover; }
.fixed-overlay { position: absolute; top: 60%; left: 50%; transform: translate(-50%, -50%); text-align: center; color: #fff; z-index: 10; background-color: rgba(0, 0, 0, 0.7); padding: 40px 60px; border-radius: 15px; box-shadow: 0px 8px 30px rgba(0, 0, 0, 0.6); width: 80%; max-width: 800px; animation: fadeIn 1.5s ease-in-out; }
.fixed-overlay h1 { font-size: 3.5rem; font-weight: bold; margin-bottom: 10px; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); }
.fixed-overlay p { font-size: 1.5rem; font-weight: normal; margin-bottom: 20px; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); }
@keyframes fadeIn { from { opacity: 0; } to { opacity: 1; } }
.featured-section { padding: 80px 0; background-color: #fff; }
.featured-section h2 { font-size: 2.5rem; font-weight: bold; margin-bottom: 40px; text-align: left; padding-left: 15px; color: #28a745; }
.feature-icon-small { display: inline-flex; align-items: center; justify-content: center; width: 70px; height: 70px; border-radius: 15px; background: linear-gradient(135deg, #28a745, #218838); margin-bottom: 25px; box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2); transition: transform 0.3s ease; }
.feature-icon-small:hover { transform: scale(1.1); }
.feature-icon-small img { width: 50px; height: 50px; }
.popular-services { padding: 80px 0; background-color: #f8f9fa; }
.popular-services h2 { font-size: 2.5rem; font-weight: bold; margin-bottom: 40px; text-align: left; padding-left: 15px; color: #28a745; }
.card-cover { background-size: cover; background-position: center; border: none; border-radius: 15px; transition: transform 0.3s ease, box-shadow 0.3s ease; overflow: hidden; }
.card-cover:hover { transform: translateY(-10px); box-shadow: 0 8px 20px rgba(0, 0, 0, 0.3); }
.card-cover h3 { font-size: 2rem; font-weight: bold; color: #fff; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); }
.card-cover p { font-size: 1.2rem; color: #fff; text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); }
.how-it-works { padding: 80px 0; background-color: #fff; }
.how-it-works h2 { font-size: 2.5rem; font-weight: bold; margin-bottom: 40px; text-align: left; padding-left: 15px; color: #28a745; }
.step-box { text-align: center; padding: 30px; border: none; border-radius: 15px; transition: transform 0.3s ease, box-shadow 0.3s ease; background-color: #fff; box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1); }
.step-box:hover { transform: translateY(-10px); box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2); }
.step-box i { font-size: 2.5rem; color: #28a745; margin-bottom: 20px; transition: color 0.3s ease; }
.step-box:hover i { color: #218838; }
.step-box h5 { font-size: 1.5rem; font-weight: bold; margin-bottom: 10px; color: #333; }
.step-box p { font-size: 1rem; color: #6c757d; }

```

```

.footer { background-color: #fff; padding: 10px 0; box-shadow: 0 -2px 10px rgba(0, 0, 0, 0.1); position: fixed; bottom: 0; left: 0; right: 0; z-index: 1000; height: 50px; display: flex; align-items: center; }
.footer .container-fluid { display: flex; justify-content: space-between; align-items: center; padding: 0 20px; }
.footer span { font-size: 0.9rem; color: #6c757d; }
.footer ul { margin: 0; padding: 0; }
.footer img { filter: grayscale(100%); transition: filter 0.3s ease; height: 20px; width: 20px; }
.footer img:hover { filter: grayscale(0%); }

css/home_base_style.css
body { background: linear-gradient(to right, #f8f9fa, #e3f2fd); font-family: 'Poppins', sans-serif; min-height: 100vh; display: flex; flex-direction: column; }
.navbar { backdrop-filter: blur(10px); box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1); }
.navbar-brand span { color: #007bff; font-weight: bold; }
.user-username { font-weight: bold; }
.service-provider-username { color: #28a745 !important; }
.customer-username { color: #007bff !important; }
.footer { background: #fff; padding: 10px 0; box-shadow: 0 -2px 10px rgba(0, 0, 0, 0.1); display: flex; justify-content: space-between; align-items: center; }
.footer img { filter: grayscale(100%); transition: filter 0.3s ease; }
.footer img:hover { filter: grayscale(0%); }

css/profile_style.css
.main-content { width: 100%; max-width: 1500px; margin: 0 auto; display: flex; flex-direction: column; gap: 40px; padding-bottom: 60px; }
.profile-section { background: #f8f9fa; border-radius: 12px; box-shadow: 0 6px 15px rgba(0, 0, 0, 0.1); padding: 30px; display: flex; align-items: center; gap: 25px; width: 100%; transition: 0.3s ease-in-out; }
.profile-section:hover { transform: scale(1.02); box-shadow: 0 8px 20px rgba(0, 0, 0, 0.15); }
.profile-large { width: 160px; height: 160px; border-radius: 50%; border: 4px solid #007bff; transition: 0.3s ease-in-out; }
.service-provider .profile-large { border-color: #28a745 !important; }
.profile-large:hover { transform: scale(1.05); box-shadow: 0 0 15px rgba(0, 123, 255, 0.6); }
.profile-info { flex: 1; text-align: left; }
.profile-info h3 { font-size: 1.8rem; font-weight: bold; color: #222; }
.profile-info p { margin: 6px 0; font-size: 1rem; color: #555; }
.notifications-section { background: #f8f9fa; border-radius: 15px; box-shadow: 0 6px 15px rgba(0, 0, 0, 0.1); padding: 25px; width: 100%; transition: 0.3s ease-in-out; }
.notifications-section:hover { transform: scale(1.02); }
.notifications-section h4 { font-size: 1.5rem; color: #FF9933; font-weight: bold; border-bottom: 3px solid #FF9933; padding-bottom: 10px; margin-bottom: 20px; }
.notifications-section p.text-muted { color: #FF9933 !important; }
.notification-list { display: flex; flex-wrap: wrap; gap: 25px; justify-content: center; }
.notification-card { flex: 1 1 calc(30% - 20px); background: #fff; border-radius: 12px; box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1); padding: 18px; transition: 0.3s ease-in-out; position: relative; overflow: hidden; text-align: left; border: 1px solid rgba(0, 0, 0, 0.1); }

```

```

.notification-card img {width:100%;height:180px;object-fit:cover;border-radius:10px;margin-bottom:12px}
.notification-card p {font-size:1rem;margin:6px 0;color:#555;text-align:left}
.notification-card p strong {font-weight:bold;color:#222}
.notification-card .status,.service-status {font-size:1.2rem;font-weight:bold;margin-top:8px;display:block;text-align:left;padding:5px 0;padding-left:10px;border-left:5px solid;background:rgba(0,123,255,0.1)}
.notification-card .status::before {content:"Status: ";font-weight:normal;color:#333}
.service-status.confirmed {border-color:#28a745;background:rgba(40,167,69,0.1);color:#218838}
.service-status.completed {border-color:#007bff;background:rgba(0,123,255,0.1);color:#0056b3}
.notification-actions {display:flex;justify-content:space-between;margin-top:15px}
.notification-actions .btn {flex:1;margin:5px;border-radius:8px;font-size:.9rem;padding:8px;transition:.3s ease-in-out}
.notification-actions .btn-success {background:#28a745;color:#fff}
.notification-actions .btn-success:hover {background:#218838}
.notification-actions .btn-danger {background:#dc3545;color:#fff}
.notification-actions .btn-danger:hover {background:#bd2130}
.notification-card:hover {transform:scale(1.05);box-shadow:0 7px 20px rgba(0,0,0,0.15)}
.booked-services-title,.booked-services-title+p {color:#FF9933}
@media(max-width:1200px){.notification-card{flex:1 1 calc(45% - 20px)}}
@media(max-width:768px){.notification-card{flex:1 1 100%}.notification-actions{flex-direction:column}.notification-actions .btn{width:100%}}
.service-provider-badge {display:inline-block;background:linear-gradient(90deg,#28a745,#20c997);color:#fff;padding:10px 18px;border-radius:25px;font-size:1.1rem;font-weight:bold;box-shadow:0 4px 10px rgba(40,167,69,0.5);transition:.2s ease-in-out;text-transform:uppercase;letter-spacing:1px}
.service-provider-badge:hover {transform:scale(1.07);box-shadow:0 5px 12px rgba(40,167,69,0.6)}
.services-section {background:#f8f9fa;border-radius:15px;box-shadow:0 6px 15px rgba(0,0,0,0.1);padding:25px;width:100%;transition:.3s ease-in-out}
.services-section:hover {transform:scale(1.02)}
.services-section h4 {font-size:1.5rem;color:#007bff;font-weight:bold;border-bottom:3px solid #007bff;padding-bottom:10px;margin-bottom:20px}
.service-list {display:flex;flex-wrap:wrap;gap:25px;justify-content:center}
.service-card {flex:1 1 calc(30% - 20px);background:#fff;border-radius:12px;box-shadow:0 5px 15px rgba(0,0,0,0.1);padding:18px;transition:.3s ease-in-out;position:relative;overflow:hidden;text-align:left;border:1px solid rgba(0,0,0,0.1)}
.service-card img {width:100%;height:180px;object-fit:cover;border-radius:10px}
.service-card h5 {font-size:1.4rem;font-weight:bold;margin-top:12px;color:#222;text-align:left}
.service-card p {font-size:1rem;margin:6px 0;color:#555;text-align:left}
.service-price {font-size:1.5rem;font-weight:bold;color:#28a745;margin-top:8px;display:block;text-align:left;padding:5px 0;padding-left:10px;border-left:5px solid #28a745;background:rgba(40,167,69,0.1)}
.service-price::before {content:"Price: ";font-weight:normal;color:#333}

```

```

.service-availability{display:inline-block;padding:6px 12px;font-size:.9rem;font-weight:bold;border-radius:6px;color:#fff;margin-top:10px;text-align:left}
.available{background:#28a745}
.not-available{background:#dc3545}
.service-actions{display:flex;justify-content:space-between;margin-top:15px}
.service-actions .btn{flex:1;margin:5px;border-radius:8px;font-size:.9rem;padding:8px;transition:.3s ease-in-out}
.service-actions .btn-primary{background:#007bff;color:#fff}
.service-actions .btn-primary:hover{background:#0056b3}
.service-actions .btn-warning{background:#ffc107;color:#000}
.service-actions .btn-warning:hover{background:#e0a800}
.service-actions .btn-danger{background:#dc3545;color:#fff}
.service-actions .btn-danger:hover{background:#bd2130}
.service-card:hover{transform:scale(1.05);box-shadow:0 7px 20px rgba(0,0,0,0.15)}
.review-card{background:#fff;padding:15px;border-radius:8px;box-shadow:0 2px 5px rgba(0,0,0,0.1);margin-bottom:15px}
.stars{font-size:1.2rem;color:#FFD700}
.star{color:#ccc}
.star.filled{color:gold}
.reviews-section{background:#fff;border-radius:12px;box-shadow:0 6px 15px rgba(0,0,0,0.1);padding:25px;width:100%}
.reviews-section h4{font-size:1.5rem;color:#28a745;font-weight:bold;border-bottom:3px solid #28a745;padding-bottom:10px;margin-bottom:20px}
.list-group-item{border-radius:8px;margin-bottom:6px;background:#f9f9f9;transition:.3s ease;padding:12px}
.list-group-item:hover{transform:translateX(8px);background:#e9f7ec}
@media(max-width:1200px){.service-card{flex:1 1 calc(45% - 20px)}}
@media(max-width:768px){
.profile-section{flex-direction:column;text-align:center}
.profile-large{width:130px;height:130px}
.service-card{flex:1 1 100%}
.service-actions{flex-direction:column}
.service-actions .btn{width:100%}
}
.service-card-footer{display:flex;justify-content:space-between;align-items:center;margin-top:15px;padding-top:10px;border-top:1px solid rgba(0,0,0,0.1)}
.service-card-footer .btn{flex:1;text-align:center;padding:10px 15px;border-radius:8px;font-size:.9rem;transition:.3s ease-in-out;margin:5px}
.payment-btn{background:#007bff;color:#fff}
.payment-btn:hover{background:#0056b3}
.cancel-btn{background:#dc3545;color:#fff}
.cancel-btn:hover{background:#bd2130}
.cancel-form{flex:1;display:flex}
.cancel-form button{width:100%}
script/index_script.js
document.addEventListener("scroll", function () {

```

```

const navbar = document.querySelector(".navbar");
if (window.scrollY > 50) {
    navbar.classList.add("navbar-scroll");
} else {
    navbar.classList.remove("navbar-scroll");
}
});
script/profile_script.js
document.addEventListener("DOMContentLoaded", function () {
    const profileImage = document.querySelector(".img-fluid");
    if (profileImage) {
        profileImage.addEventListener("mouseenter", function () { this.style.transform =
"scale(1.1)"; this.style.transition = "0.3s ease-in-out"; });
        profileImage.addEventListener("mouseleave", function () { this.style.transform =
"scale(1)"; });
    }
    const serviceCards = document.querySelectorAll(".service-card");
    serviceCards.forEach(card => {
        card.addEventListener("mouseenter", function () { this.style.boxShadow = "0px 5px 20px
rgba(0, 120, 232, 0.3)"; this.style.transition = "0.3s"; });
        card.addEventListener("mouseleave", function () { this.style.boxShadow = "none"; });
    });
    const reviewLink = document.getElementById("reviews-link");
    if (reviewLink) {
        reviewLink.addEventListener("click", function (event) { event.preventDefault();
document.getElementById("reviews-section").scrollIntoView({ behavior: "smooth" }); });
    }
    const logoutButton = document.querySelector(".logout-button");
    if (logoutButton) {
        logoutButton.addEventListener("click", function (event) { event.preventDefault(); if
(confirm("Are you sure you want to log out?")) { window.location.href = this.getAttribute("data-
url"); } });
    }
});

```

**templates****base.html**

```

{%- load static %}

<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>{{ block title }}</title>
        <link rel="icon" href="{{ static 'appicon.ico' }}" type="image/icon">

```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
  {% block style %} {% endblock %}
</head>
<body>
  {% block content %} {% endblock %}
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvperYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
  {% block script %} {% endblock %}
  <script></script>
</body>
</html>
home_base.html
{% load static %}
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>{% block title %} {% endblock %}</title>
  <link rel="icon" href="{% static 'appicon.ico' %}" type="image/icon">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
  <link rel="stylesheet" href="{% static 'css/home_base_style.css' %}">
  {% block style %} {% endblock %}
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-white shadow-sm fixed-top navbar-
scroll">
    <div class="container-fluid d-flex justify-content-between align-items-center">
      <a class="navbar-brand d-flex align-items-center" href="{% url 'profile' %}">
        
        <span class="fw-bold">EduMart</span>
      </a>
      <form class="d-flex mx-auto w-50" action="{% url 'search_services' %}" method="GET">
        <input class="form-control me-2" type="search" placeholder="Search by title, category, or
provider..." aria-label="Search" name="q" value="{{ request.GET.q }}"/>
        <button class="btn btn-outline-primary" type="submit">Search</button>
      </form>
    <div class="d-flex align-items-center">

```

```

<button class="btn btn-outline-primary me-2 fw-normal"
onclick="window.location.href='{{ url 'service_list' }}'">Browse Services</button>
    {% if user.is_service_provider %}
        <button class="btn btn-outline-primary me-3 fw-normal"
onclick="window.location.href='{{ url 'create_service' }}'">Add New Service</button>
    {% endif %}
    <a href="{{ url 'profile' }}" class="user-profile d-flex align-items-center text-decoration-none">
        <span class="fst-italic me-2 m-0 p-0 user-username {{ user.is_service_provider
}}service-provider-username{{ else }}customer-username{{ endif }}">{{ user.username
}}</span>
        {% if user.profile_picture %}
            
        {% endif %}
    </a>
    <a href="{{ url 'logout' }}" class="btn btn-dark ms-3 fw-normal">Logout</a>
</div>
</div>
</nav>
{% block content %}{% endblock %}
<footer class="footer fixed-bottom">
    <div class="container-fluid d-flex justify-content-between align-items-center">
        <span class="text-body-secondary">© {{ now | Y }} EduMart</span>
        <ul class="nav list-unstyled d-flex">
            <li class="ms-3"><a class="text-body-secondary" href="#"></a></li>
            <li class="ms-3"><a class="text-body-secondary" href="#"></a></li>
            <li class="ms-3"><a class="text-body-secondary" href="#"></a></li>
        </ul>
    </div>
</footer>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
    {% block script %}{% endblock %}
</body>
</html>
index.html
{% extends "base.html" %}
{% load static %}
{% block title %} EduMart | Marketplace for Educational Services {% endblock %}

```

```

{% block style %}<link rel="stylesheet" href="{% static 'css/index_styles.css' %}">{%
endblock %}
{% block content %}
<nav class="navbar navbar-expand-lg navbar-light bg-white shadow-sm fixed-top navbar-
scroll">
<div class="container-fluid d-flex justify-content-between align-items-center">
<a class="navbar-brand" href="{% url 'index' %}"> EduMart</a>
<div class="flex-grow-1 text-center"><p class="mb-0 fst-italic"> 🔎 Find the Right Service for
You!</p></div>
<div class="ms-auto"><button class="btn btn-primary me-2" type="button"
onclick="window.location.href='{{ url 'login' }}'">Login</button><button class="btn btn-
outline-primary" type="button" onclick="window.location.href='{{ url 'register' }}'">Register</button></div>
</div>
</nav>
<div id="carouselExampleCaptions" class="carousel slide">
<div class="carousel-indicators">
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide 1"></button>
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-
label="Slide 2"></button>
<button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-
label="Slide 3"></button>
</div>
<div class="carousel-inner">
<div class="carousel-item active"></div>
<div class="carousel-item"></div>
<div class="carousel-item"></div>
</div>
<div class="fixed-overlay"><h1>Welcome to EduMart - Your Marketplace for Educational
Services</h1><p>Find expert tutors, designers, developers, and more – all in one
place!</p></div>
<button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="prev"><span class="carousel-control-prev-
icon" aria-hidden="true"></span><span class="visually-hidden">Previous</span></button>
<button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="next"><span class="carousel-control-next-
icon" aria-hidden="true"></span><span class="visually-hidden">Next</span></button>
</div>
<div class="container-fluid px-4 py-5">
<h2 class="pb-2 border-bottom text-center">Why Choose EduMart?</h2>
<div class="row row-cols-1 row-cols-md-2 align-items-md-center g-5 py-5">

```

```

<div class="col d-flex flex-column align-items-start gap-2">
  <h2 class="fw-bold text-body-emphasis">Learn, Grow, and Hire with Confidence!</h2>
  <p class="text-body-secondary">EduMart is your one-stop platform for finding trusted experts in tutoring, graphic design, and app development. With easy booking, secure payments, and real user reviews, we ensure a smooth and reliable experience for both clients and service providers.
    <img alt="rocket icon" data-bbox="118 181 181 198" style="vertical-align: middle;"/>
  </p>
</div>
<div class="col">
  <div class="row row-cols-1 row-cols-sm-2 g-4">
    <div class="col d-flex flex-column gap-2">
      <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3"></div>
      <h4 class="fw-semibold mb-0 text-body-emphasis">Trusted Professionals</h4><p class="text-body-secondary">Verified tutors & experts.</p>
    </div>
    <div class="col d-flex flex-column gap-2">
      <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3"></div>
      <h4 class="fw-semibold mb-0 text-body-emphasis">Easy Booking</h4><p class="text-body-secondary">Schedule sessions at your convenience.</p>
    </div>
    <div class="col d-flex flex-column gap-2">
      <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3"></div>
      <h4 class="fw-semibold mb-0 text-body-emphasis">Secure Payments</h4><p class="text-body-secondary">Pay safely with integrated gateways.</p>
    </div>
    <div class="col d-flex flex-column gap-2">
      <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3"></div>
      <h4 class="fw-semibold mb-0 text-body-emphasis">Real Reviews</h4><p class="text-body-secondary">Ratings & feedback from real users.</p>
    </div>
    </div>
  </div>
  </div>
</div>
<div class="container-fluid px-4 py-5" id="custom-cards">
  <h2 class="pb-2 border-bottom text-center">Popular Services</h2>
  <div class="row row-cols-1 row-cols-lg-3 align-items-stretch g-4 py-5">
    <div class="col">

```

```

<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg"
style="background-image: url('% static 'images/tutoring.png' %');">
<div class="d-flex flex-column h-100 p-5 pb-3 text-white text-shadow-1">
<h3 class="pt-5 mt-5 mb-4 display-6 lh-1 fw-bold">Tutoring</h3><p class="fs-5 mb-4">Math,
Science, Coding, and more.</p>
</div>
</div>
</div>
<div class="col">
<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg"
style="background-image: url('% static 'images/graphics-design.png' %');">
<div class="d-flex flex-column h-100 p-5 pb-3 text-white text-shadow-1">
<h3 class="pt-5 mt-5 mb-4 display-6 lh-1 fw-bold">Graphic Design</h3><p class="fs-5 mb-4">Logos, UI/UX Illustrations.</p>
</div>
</div>
</div>
<div class="col">
<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg"
style="background-image: url('% static 'images/app-dv.png' %');">
<div class="d-flex flex-column h-100 p-5 pb-3 text-shadow-1">
<h3 class="pt-5 mt-5 mb-4 display-6 lh-1 fw-bold">App Development</h3><p class="fs-5 mb-4">Websites, mobile apps, software.</p>
</div>
</div>
</div>
</div>
<section class="container text-center mt-5">
<h2> How It Works?</h2>
<p class="text-muted">A simple 4-step process to connect with the best educational services.</p>
<div class="row mt-4 how-it-works">
<div class="col-md-3">
<div class="step-box"><i class="bi bi-person-plus"></i><h5>Sign Up</h5><p>Create your free account</p></div>
</div>
<div class="col-md-3">
<div class="step-box"><i class="bi bi-search"></i><h5>Explore Services</h5><p>Find what you need</p></div>
</div>
<div class="col-md-3">
<div class="step-box"><i class="bi bi-credit-card"></i><h5>Book & Pay</h5><p>Secure your booking</p></div>
</div>
<div class="col-md-3">

```

```

<div class="step-box"><i class="bi bi-lightbulb"></i><h5>Learn & Grow</h5><p>Get expert
help!</p></div>
</div>
</div>
</section>
<footer class="footer fixed-bottom">
<div class="container-fluid d-flex justify-content-between align-items-center">
<span class="text-body-secondary">© {% now 'Y' %} EduMart</span>
<ul class="nav list-unstyled d-flex">
<li class="ms-3"><a class="text-body-secondary" href="#"></a></li>
<li class="ms-3"><a class="text-body-secondary" href="#"></a></li>
<li class="ms-3"><a class="text-body-secondary" href="#"></a></li>
</ul>
</div>
</footer>
{% endblock %}
{% block script %}<script src="{% static 'script/index_script.js' %}"></script>{% endblock %}
profile.html
{% extends "home_base.html" %}
{% load static %}
{% block title %}Profile | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/profile_style.css' %}">
{% endblock %}
{% block content %}
<div class="container mt-5 pt-5">
<div class="main-content">
{% if user.is_authenticated %}
    {% include 'partials/profile_section.html' %}
    {% if user.is_service_provider %}
        {% include 'partials/booking_requests.html' %}
        {% include 'partials/booked_services_provider.html' %}
        {% include 'partials/completed_services_provider.html' %}
        {% include 'partials/your_services.html' %}
        {% include 'partials/reviews_provider.html' %}
    {% else %}
        {% include 'partials/your_bookings.html' %}
        {% include 'partials/booked_services_client.html' %}
        {% include 'partials/completed_services_client.html' %}
        {% include 'partials/reviews_client.html' %}
    {% endif %}
    {% endif %}
</div>
</div>

```

```

</div>
{% endblock %}
{% block script %}
<script src="{% static 'script/profile_script.js' %}" defer></script>
{% endblock %}
partials/booked_services_client.html
<div class="services-section">
  <h4>Your Confirmed Bookings</h4>
  <div class="service-list">
    {% for booking in client_bookings %}
      {% include 'partials/service_card.html' with booking=booking %}
    {% empty %}
      <p class="text-muted">No confirmed bookings yet.</p>
    {% endfor %}
  </div>
</div>
partials/booked_services_provider.html
<div class="services-section">
  <h4>Services You Have Booked</h4>
  <div class="service-list">
    {% for booking in provider_bookings %}
      {% include 'partials/service_card.html' with booking=booking %}
    {% empty %}
      <p class="text-muted">No confirmed bookings yet.</p>
    {% endfor %}
  </div>
</div>
partials/booked_services.html
<div class="services-section">
  <h4>Booked Services</h4>
  <div class="service-list">
    {% for booking in bookings %}
      {% if booking.status != "Cancelled" and booking.status != "Completed" %}
        {% include 'partials/service_card.html' with booking=booking %}
      {% endif %}
    {% empty %}
      <p class="text-muted">No booked services.</p>
    {% endfor %}
  </div>
</div>
partials/booking_requests.html
<div class="notifications-section">
  <h4>Booking Requests</h4>
  {% if notifications %}
    <div class="notification-list">
      {% for booking in notifications %}

```

```

<div class="notification-card">
    {% if booking.service.image %}
        
    {% endif %}
    <p><strong>Client:</strong> {{ booking.client.username }}</p>
    <p><strong>Service:</strong> {{ booking.service.title }}</p>
    <p><strong>Category:</strong> {{ booking.service.category }}</p>
    <p><strong>Price:</strong> ₹{{ booking.service.price }}</p>
    <p><strong>Requested Date:</strong> {{ booking.date }}</p>
    <p><strong>Status:</strong> {{ booking.status }}</p>
    <div class="notification-actions">
        <a href="{% url 'accept_booking' booking.id %}" class="btn btn-success">Accept</a>
        <a href="{% url 'reject_booking' booking.id %}" class="btn btn-danger">Reject</a>
    </div>
</div>
    {% endfor %}
</div>
    {% else %}
        <p class="text-muted">No new booking requests.</p>
    {% endif %}
</div>

```

**completed\_services\_client.html**

```

<div class="services-section">
    <h4>Completed Services</h4>
    <div class="service-list">
        {% for booking in client_completed_services %}
            {% include 'partials/service_card.html' with booking=booking %}
        {% empty %}
            <p class="text-muted">No completed services yet.</p>
        {% endfor %}
    </div>
</div>

```

**completed\_services\_provider.html**

```

<div class="services-section">
    <h4>Your Completed Services</h4>
    <div class="service-list">
        {% for booking in provider_completed_services %}
            {% include 'partials/service_card.html' with booking=booking %}
        {% empty %}
            <p class="text-muted">No completed services yet.</p>
        {% endfor %}
    </div>
</div>

```

**completed\_services.html**

```
<div class="services-section">
    <h4>Completed Services</h4>
    <div class="service-list">
        {% for booking in bookings %}
            {% if booking.status == "Completed" %}
                {% include 'partials/service_card.html' with booking=booking %}
            {% endif %}
            {% empty %}
                <p class="text-muted">No completed services.</p>
            {% endfor %}
    </div>
</div>
```

**profile\_section.html**

```
<div class="profile-section {% if user.is_service_provider %}service-provider{% endif %}">
    
    <div class="profile-info">
        {% if user.is_service_provider %}
            <span class="service-provider-badge">Service Provider</span>
        {% endif %}
        <h3>{{ user.username }}</h3>
        <p class="text-muted"><strong>Email:</strong> {{ user.email }}</p>
        <p><strong>Bio:</strong> {{ user.bio }}</p>
        <a href="{% url 'edit_profile' %}" class="btn btn-warning mt-3">Edit Profile</a>
    </div>
</div>
```

**reviews\_client.html**

```
<div class="reviews-section">
    <h4>Your Reviews</h4>
    <div class="review-list">
        {% if client_reviews %}
            {% for review in client_reviews %}
                <div class="review-card">
                    <p><strong>Service:</strong> {{ review.service.title }}</p>
                    <p><strong>Provider:</strong> {{ review.service.provider.username }}</p>
                    <p><strong>Reviewer:</strong> {{ review.client.username }}</p>
                    <p><strong>Rating:</strong>
                        <span class="stars">
                            {% for i in "12345" | make_list %}
                                {% if forloop.counter <= review.rating %}
                                    <span class="star filled">★</span>
                                {% else %}
                                    <span class="star">☆</span>
                                {% endif %}
                            {% endfor %}
                        </span>
                
```

```

        </span> ({{ review.rating }}/5)
    </p>
    <p class="review-comment"><strong>Comment:</strong> {{ review.comment
}}</p>
    <p class="review-date text-muted"><strong>Reviewed on:</strong> {{
review.created_at|date:"M d, Y" }} at {{ review.created_at|time:"h:i A" }}</p>
    {% if review.client == request.user %}
        <div class="review-actions">
            <a href="{% url 'edit_review' review.id %}" class="btn btn-warning btn-
sm">Edit</a>
            <a href="{% url 'delete_review' review.id %}" class="btn btn-danger btn-sm"
onclick="return confirm('Are you sure you want to delete this review?');">Delete</a>
        </div>
    {% endif %}
    </div>
    {% endfor %}
    {% else %}
        <p class="text-muted">No reviews yet.</p>
    {% endif %}
</div>
</div>
reviews_provider.html
<div class="reviews-section">
    <h4>Reviews for Your Services</h4>
    <div class="review-list">
        {% if provider_reviews %}
            {% for review in provider_reviews %}
                <div class="review-card">
                    <p><strong>Service:</strong> {{ review.service.title }}</p>
                    <p><strong>Provider:</strong> {{ review.service.provider.username }}</p>
                    <p><strong>Reviewer:</strong> {{ review.client.username }}</p>
                    <p><strong>Rating:</strong>
                        <span class="stars">
                            {% for i in "12345"|make_list %}
                                {% if forloop.counter <= review.rating %}
                                    <span class="star filled">★</span>
                                {% else %}
                                    <span class="star">☆</span>
                                {% endif %}
                            {% endfor %}
                        </span> ({{ review.rating }}/5)
                    </p>
                    <p class="review-comment"><strong>Comment:</strong> {{ review.comment
}}</p>
                    <p class="review-date text-muted"><strong>Reviewed on:</strong> {{
review.created_at|date:"M d, Y" }} at {{ review.created_at|time:"h:i A" }}</p>
                </div>
            {% endfor %}
        {% endif %}
    </div>
</div>

```

```

        </div>
    {% endfor %}
    {% else %}
        <p class="text-muted">No reviews yet.</p>
    {% endif %}
</div>
</div>
review_section.html
<div class="reviews-section">
    <h4>Reviews & Ratings</h4>
    <div class="review-list">
        {% for review in reviews %}
            <div class="review-card">
                <p><strong>Service:</strong> {{ review.service.title }}</p>
                <p><strong>Reviewer:</strong> {{ review.user.username }}</p>
                <p><strong>Rating:</strong> ★ {{ review.rating }}/5</p>
                <p class="review-comment"><strong>Comment:</strong> {{ review.comment }}</p>
            </div>
            <p class="review-date text-muted">Reviewed on: {{ review.created_at|date:"M d, Y" }}</p>
        {% empty %}
            <p class="text-muted booked-services-title">No reviews yet.</p>
        {% endfor %}
    </div>
</div>
service_card.html
<div class="service-card">
    <div class="service-card-header">
        {% if booking.service.image %}
            
        {% endif %}
        <div class="service-info">
            <h5>{{ booking.service.title }}</h5>
            <p class="provider"><strong>Provider:</strong> {{ booking.service.provider.username }}</p>
            <p class="client"><strong>Client:</strong> {{ booking.client.username }}</p>
        </div>
    </div>
    <div class="service-card-body">
        <p><strong>Price:</strong> ₹{{ booking.service.price }}</p>
        <p><strong>Booked On:</strong> {{ booking.created_at|date:"d M Y, H:i A" }}</p>
        <p><strong>Service Date:</strong> {{ booking.date }}</p>
        <p><strong>Service Time:</strong> {{ booking.time }}</p>
    </div>
</div>

```

```

<p><strong>Status:</strong> <span class="status-{{ booking.status|lower }}">{{ booking.status }}</span></p>
</div>
<div class="service-card-footer">
  {% if booking.status == "Approved" %}
    <a href="{% url 'make_payment' booking.id %}" class="btn btn-primary payment-btn">Make Payment</a>
  {% endif %}
  {% if booking.status != "Confirmed" and booking.status != "Completed" %}
    <form action="{% url 'cancel_booking' booking.id %}" method="post" class="inline-form cancel-form">
      {% csrf_token %}
      <button type="submit" class="btn btn-danger cancel-btn">Cancel</button>
    </form>
  {% endif %}
  {% if booking.status == "Confirmed" %}
    <a href="{% url 'chat_room' booking.id %}" class="btn btn-success chat-btn">Chat</a>
  {% endif %}
  {% if booking.status == "Completed" and request.user == booking.client %}
    <a href="{% url 'add_review' booking.service.id %}" class="btn btn-warning review-btn">Add Review</a>
  {% endif %}
  {% if booking.status == "Completed" and request.user == booking.service.provider %}
    <a href="{% url 'view_service_reviews' booking.service.id %}" class="btn btn-info review-btn">View Reviews</a>
  {% endif %}
</div>
</div>
your_bookings.html
<div class="services-section">
  <h4>Your Bookings</h4>
  {% if bookings_client.exists %}
    <div class="service-list">
      {% for booking in bookings_client %}
        {% include 'partials/service_card.html' with booking=booking %}
      {% endfor %}
    </div>
  {% else %}
    <p class="text-muted no-services">No booked services yet.</p>
  {% endif %}
</div>
your_services.html
<div class="services-section">
  <h4>Your Services</h4>
  <div class="service-list">
    {% for service in services %}

```

```

<div class="service-card">
    {% if service.image %}
        
    {% endif %}
    <h5>{{ service.title }}</h5>
    <p><strong>Category:</strong> {{ service.category }}</p>
    <p><strong>Price:</strong> ₹{{ service.price }}</p>
    <p><strong>Availability:</strong> {{% if service.available %}}  Available {{% else %}}  Not Available {{% endif %}}</p>
    <p><strong>Created on:</strong> {{ service.created_at|date:"M d, Y" }}</p>
    <p class="service-desc"><strong>Description:</strong> {{ service.description }}</p>
    <div class="service-actions">
        <a href="{% url 'update_service' service.id %}" class="btn btn-warning">Edit</a>
        <a href="{% url 'delete_service' service.id %}" class="btn btn-danger"
        onclick="return confirm('Are you sure you want to delete this service?');">Delete</a>
        <a href="{% url 'view_service_reviews' service.id %}" class="btn btn-info">View
    Reviews</a>
    </div>
    </div>
    {% empty %}
    <p class="text-muted">You haven't listed any services yet.</p>
    {% endfor %}
</div>
</div>

```

**users****admin.py**

```

from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import CustomUser
class CustomUserAdmin(UserAdmin):
    model = CustomUser
    list_display = ('username', 'email', 'is_service_provider', 'is_staff', 'is_active', 'date_joined')
    search_fields = ('username', 'email')
    list_filter = ('is_service_provider', 'is_staff', 'is_active', 'date_joined')
    fieldsets = UserAdmin.fieldsets + ((("Additional Info", {"fields": ("is_service_provider",
    "profile_picture", "bio")}),)
    add_fieldsets = UserAdmin.add_fieldsets + ((("Additional Info", {"fields": ("is_service_provider", "profile_picture", "bio")}),))
admin.site.register(CustomUser, CustomUserAdmin)

```

**apps.py**

```

from django.apps import AppConfig
class UsersConfig(AppConfig):
    default_auto_field = "django.db.models.BigAutoField"
    name = "users"

```

**models.py**

```

from django.contrib.auth.models import AbstractUser

```

```

from django.db import models
class CustomUser(AbstractUser):
    is_service_provider = models.BooleanField(default=False)
    profile_picture = models.ImageField(upload_to='profile_pics/', blank=True, null=True,
default='profile_pics/default.png')
    bio = models.TextField(blank=True, null=True)
    def __str__(self):
        return f'{self.username} ({self.email})'

urls.py
from django.urls import path
from django.conf import settings
from django.conf.urls.static import static
from .views import register, verify_otp, resend_otp, user_login, user_logout,
request_password_reset, verify_password_otp, reset_password, edit_profile
urlpatterns = [
    path("register/", register, name="register"),
    path("edit-profile/", edit_profile, name="edit_profile"),
    path("verify-otp/", verify_otp, name="verify_otp"),
    path("resend-otp/", resend_otp, name="resend_otp"),
    path("login/", user_login, name="login"),
    path("logout/", user_logout, name="logout"),
    path("password-reset/", request_password_reset, name="request_password_reset"),
    path("verify-password-otp/", verify_password_otp, name="verify_password_otp"),
    path("reset-password/", reset_password, name="reset_password"),
]
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

views.py
import os
from django.shortcuts import render, redirect
from django.contrib.auth import login, logout, authenticate
from django.contrib import messages
from django.core.mail import send_mail
from django.conf import settings
from django.contrib.auth.hashers import make_password
from django.contrib.auth.decorators import login_required
from django.utils.crypto import get_random_string
from users.models import CustomUser
def generate_otp():
    return get_random_string(6, allowed_chars='0123456789')
def send_otp_email(email, otp, subject):
    message = f"Your OTP is: {otp}. Do not share this with anyone."
    send_mail(subject, message, settings.EMAIL_HOST_USER, [email], fail_silently=False)
def register(request):
    if request.method == "POST":
        username = request.POST.get("username")

```

```

email = request.POST.get("email")
password1 = request.POST.get("password1")
password2 = request.POST.get("password2")
is_service_provider = request.POST.get("is_service_provider") == "on"
bio = request.POST.get("bio", "")
profile_picture = request.FILES.get("profile_picture")
if password1 != password2:
    messages.error(request, "Passwords do not match.")
    return redirect("register")
if CustomUser.objects.filter(email=email).exists():
    messages.error(request, "A user with this email already exists.")
    return redirect("register")
temp_profile_pic = None
if profile_picture:
    temp_path = os.path.join(settings.MEDIA_ROOT, 'temp', profile_picture.name)
    os.makedirs(os.path.dirname(temp_path), exist_ok=True)
    with open(temp_path, 'wb+') as destination:
        for chunk in profile_picture.chunks():
            destination.write(chunk)
    temp_profile_pic = os.path.join('temp', profile_picture.name)
otp = generate_otp()
request.session["pending_registration"] = {
    "username": username,
    "email": email,
    "password": make_password(password1),
    "is_service_provider": is_service_provider,
    "bio": bio,
    "temp_profile_pic": temp_profile_pic,
}
request.session["registration_otp"] = otp
request.session.set_expiry(600)
send_otp_email(email, otp, "EduMart OTP Verification")
messages.success(request, "OTP sent to your email. Please verify.")
return redirect("verify_otp")
return render(request, "users/register.html")
def verify_otp(request):
    if request.method == "POST":
        entered_otp = request.POST.get("otp")
        session_otp = request.session.get("registration_otp")
        user_data = request.session.get("pending_registration")
        if user_data and entered_otp == session_otp:
            temp_profile_pic = user_data.pop("temp_profile_pic", None)
            user = CustomUser.objects.create(**user_data)
            if temp_profile_pic:
                temp_path = os.path.join(settings.MEDIA_ROOT, temp_profile_pic)
                if os.path.exists(temp_path):

```

```

        with open(temp_path, 'rb') as f:
            user.profile_picture.save(os.path.basename(temp_profile_pic), f, save=True)
            os.remove(temp_path)
    del request.session["registration_otp"]
    del request.session["pending_registration"]
    messages.success(request, "Registration successful! You can now log in.")
    return redirect("login")
else:
    messages.error(request, "Invalid OTP. Please try again.")
    return render(request, "users/verify_otp.html")
def resend_otp(request):
    user_email = request.session.get("pending_registration", {}).get("email")
    if user_email:
        new_otp = generate_otp()
        request.session["registration_otp"] = new_otp
        send_otp_email(user_email, new_otp, "EduMart OTP Resend")
        messages.success(request, "A new OTP has been sent to your email.")
        return redirect("verify_otp")
    messages.error(request, "Session expired. Please register again.")
    return redirect("register")
def user_login(request):
    if request.method == "POST":
        username = request.POST.get("username")
        password = request.POST.get("password")
        user = authenticate(request, username=username, password=password)
        if user:
            login(request, user)
            return redirect("profile")
        else:
            messages.error(request, "Invalid credentials. Please try again.")
            return render(request, "users/login.html")
@login_required
def edit_profile(request):
    user = request.user
    if request.method == "POST":
        username = request.POST.get("username")
        bio = request.POST.get("bio")
        profile_picture = request.FILES.get("profile_picture")
        user.username = username
        user.bio = bio
        if profile_picture:
            user.profile_picture = profile_picture
        user.save()
        messages.success(request, "Profile updated successfully!")
        return redirect("profile")
    return render(request, "users/edit_profile.html", {"user": user})

```

```

def user_logout(request):
    logout(request)
    return redirect("index")
def request_password_reset(request):
    if request.method == "POST":
        username = request.POST.get("username")
        email = request.POST.get("email")
        try:
            user = CustomUser.objects.get(username=username, email=email)
            otp = generate_otp()
            request.session["password_reset_otp"] = otp
            request.session["reset_user"] = user.email
            request.session.set_expiry(600)
            send_otp_email(email, otp, "EduMart Password Reset OTP")
            messages.success(request, "OTP sent to your email.")
            return redirect("verify_password_otp")
        except CustomUser.DoesNotExist:
            messages.error(request, "Invalid username or email.")
        return render(request, "users/request_password_reset.html")
    def verify_password_otp(request):
        if request.method == "POST":
            entered_otp = request.POST.get("otp")
            if entered_otp == request.session.get("password_reset_otp"):
                return redirect("reset_password")
            messages.error(request, "Invalid OTP.")
        return render(request, "users/verify_password_otp.html")
    def reset_password(request):
        if request.method == "POST":
            new_password = request.POST.get("new_password")
            confirm_password = request.POST.get("confirm_password")
            if new_password == confirm_password:
                user = CustomUser.objects.get(email=request.session.get("reset_user"))
                user.password = make_password(new_password)
                user.save()
                del request.session["password_reset_otp"]
                del request.session["reset_user"]
                messages.success(request, "Password reset successfully.")
                return redirect("login")
            messages.error(request, "Passwords do not match.")
        return render(request, "users/reset_password.html")
    static

```

**users/css/base\_style.css**

```

body {background-color: #f8f9fa; font-family: 'Poppins', sans-serif; display: flex; justify-content: center; align-items: center; min-height: 100vh; margin: 0; overflow: auto;}
html {scroll-behavior: smooth; height: 100%;}

```

```
.form-container {width: 90%; max-width: 500px; background: #ffffff; padding: 25px; border-radius: 12px; box-shadow: 0px 4px 12px rgba(0, 0, 0, 0.15); text-align: center; border: 2px solid #e0e0e0;}
.brand-container {display: flex; align-items: center; justify-content: center; gap: 12px; margin-bottom: 20px;}
.brand-container img {width: 50px; height: auto;}
.brand-container h1 {font-size: 32px; font-weight: 700; color: #0078E8; margin: 0;}
h2 {font-size: 18px; font-weight: 700; margin-bottom: 20px; color: #0078E8;}
p {font-size: 14px; color: #6c757d; margin-bottom: 15px;}
.form-group {margin-bottom: 20px;}
.form-floating {width: 100%;}
.form-floating input {width: 100%; padding: 12px; border-radius: 6px; border: 1px solid #0078E8; transition: all 0.3s ease;}
.form-floating input:focus {border-color: #0056B3; box-shadow: 0px 0px 6px rgba(0, 123, 255, 0.5);}
.form-floating label {font-size: 14px; color: #000; margin-bottom: 6px; display: block;}
.btn-primary, .btn-success {width: 100%; background-color: #0078E8; border: none; border-radius: 8px; padding: 12px; font-size: 16px; font-weight: 600; transition: all 0.3s ease-in-out;}
.btn-primary:hover, .btn-success:hover {background-color: #0056B3;}
.login-links a, .login-link a {text-decoration: none; color: #0078E8; font-weight: 600;}
.login-links a:hover, .login-link a:hover {text-decoration: underline; color: #0056B3;}
.footer {margin-top: 20px; font-size: 14px; color: #6c757d;}
@media (max-width: 480px) {.form-container {padding: 20px;} .btn-primary, .btn-success {font-size: 14px;}}
users/css/login_style.css
@import url("base_style.css");
.form-check {display: flex; align-items: center; gap: 8px; justify-content: flex-start; text-align: left;}
users/css/password_reset_style.css
@import url("base_style.css");
.btn-secondary {width: 100%; background-color: #6c757d; border: none; border-radius: 8px; padding: 12px; font-size: 16px; font-weight: 600; transition: all 0.3s ease-in-out;}
.btn-secondary:hover {background-color: #5a6268;}
users/css/register_style.css
@import url("base_style.css");
.form-group {display: flex; justify-content: space-between; gap: 20px; margin-bottom: 20px;}
.form-check {display: flex; align-items: center; gap: 8px;}
.form-check-label {margin: 0; text-align: left;}
users/css/reset_password_style.css
@import url("base_style.css");
users/css/verify_otp_style.css
@import url("base_style.css");
users/css/verify_password_otp_style.css
@import url('base_style.css');
```

**templates****users/register.html**

```

{% extends "base.html" %}
{% load static %}
{% block title %}Register | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'users/css/register_style.css' %}">
{% endblock %}
{% block content %}
<div class="form-container">
    <form method="POST" action="{% url 'register' %}" enctype="multipart/form-data">
        {% csrf_token %}
        <div class="brand-container">
            <a href="{% url 'index' %}" class="logo-button">
                
            </a>
            <h1>EduMart</h1>
        </div>
        <h2 class="text-center">Please Register Yourself</h2>
        {% if messages %}
            <div class="alert alert-danger">
                {% for message in messages %}
                    <p>{{ message }}</p>
                {% endfor %}
            </div>
        {% endif %}
        <div class="form-group">
            <div class="form-floating mb-3">
                <input type="email" class="form-control" name="email" id="email"
placeholder="Email" required>
                <label for="email">Email</label>
            </div>
            <div class="form-floating mb-3">
                <input type="text" class="form-control" name="username" id="username"
placeholder="Username" required>
                <label for="username">Username</label>
            </div>
        </div>
        <div class="form-group">
            <div class="form-floating mb-3">
                <input type="password" class="form-control" name="password1" id="password1"
placeholder="Password" required>
                <label for="password1">Password</label>
            </div>
            <div class="form-floating mb-3">
                <input type="password" class="form-control" name="password2" id="password2"
placeholder="Confirm Password" required>
                <label for="password2">Confirm Password</label>
            </div>
        </div>
    </form>
</div>

```

```

        </div>
    </div>
    <div class="form-group profile-picture-container">
        <label for="profile_picture">Profile Picture</label>
        <input type="file" class="form-control" name="profile_picture" id="profile_picture">
    </div>
    <div class="form-group mb-3">
        <div class="form-floating">
            <textarea class="form-control" name="bio" id="bio" placeholder="Bio"></textarea>
            <label for="bio">Bio</label>
        </div>
    </div>
    <div class="form-check mb-3">
        <input type="checkbox" class="form-check-input" name="is_service_provider"
id="is_service_provider">
        <label class="form-check-label" for="is_service_provider">I am a Service
Provider</label>
    </div>
    <button class="btn btn-primary w-100" type="submit">Register</button>
    <div class="login-link mt-3">
        <a href="{% url 'login' %}">Already have an account? Login</a>
    </div>
</form>
</div>
{% endblock %}
users/login.html
{% extends "base.html" %}
{% load static %}
{% block title %}Login | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'users/css/base_style.css' %}">
<link rel="stylesheet" href="{% static 'users/css/login_style.css' %}">
{% endblock %}
{% block content %}
<div class="form-container">
    <form method="POST" action="{% url 'login' %}">
        {% csrf_token %}
        <div class="brand-container">
            <a href="{% url 'index' %}" class="logo-button">
                
            </a>
            <h1>EduMart</h1>
        </div>
        <h2 class="text-center">Please Login Yourself</h2>
        {% if messages %}
        <div class="alert alert-danger">

```

```

    {% for message in messages %}
        <p>{{ message }}</p>
    {% endfor %}
</div>
{% endif %}
<div class="form-group mb-3">
    <div class="form-floating">
        <input type="text" class="form-control" name="username" id="username"
placeholder="Username" required>
            <label for="username"><i class="fas fa-user"></i> Username</label>
        </div>
    </div>
    <div class="form-group mb-3">
        <div class="form-floating">
            <input type="password" class="form-control" name="password" id="password"
placeholder="Password" required>
                <label for="password"><i class="fas fa-lock"></i> Password</label>
            </div>
        </div>
        <div class="form-check mb-3">
            <input type="checkbox" class="form-check-input" id="remember_me"
name="remember_me">
                <label class="form-check-label" for="remember_me">Remember Me</label>
            </div>
        <button class="btn btn-primary w-100" type="submit">Login</button>
        <div class="login-links mt-3">
            <a href="{% url 'register' %}">Create new account</a> |
            <a href="{% url 'request_password_reset' %}">Forgot Password?</a>
        </div>
    </form>
</div>
{% endblock %}
users/request_password_reset.html
{% extends "base.html" %}
{% load static %}
{% block title %} Request Password Reset | EduMart {% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/base_style.css' %}">
<link rel="stylesheet" href="{% static 'users/css/password_reset_style.css' %}">
{% endblock %}
{% block content %}
<div class="form-container">
    <form method="POST" action="{% url 'request_password_reset' %}"
onsubmit="disableSubmitButton()">
        {% csrf_token %}
        <div class="brand-container">

```

```

<a href="{% url 'index' %}>
    
</a>
<h1>EduMart</h1>
</div>
<h2>Request Password Reset</h2>
<p>Enter your username and email to receive an OTP for password reset.</p>
{% if messages %}
<div class="alert alert-danger">
    {% for message in messages %}
        <p>{{ message }}</p>
    {% endfor %}
</div>
{% endif %}
<div class="form-group">
    <div class="form-floating">
        <input type="text" class="form-control" name="username" id="username"
placeholder="Username" required>
            <label for="username">Username</label>
        </div>
    </div>
    <div class="form-group">
        <div class="form-floating">
            <input type="email" class="form-control" name="email" id="email"
placeholder="Email" required>
                <label for="email">Email</label>
            </div>
        </div>
        <button id="submitBtn" class="btn btn-primary w-100" type="submit">Send
OTP</button>
        <a href="{% url 'login' %}" class="btn btn-secondary w-100 mt-2">Go Back</a>
    </form>
</div>
{% endblock %}
{% block script %}
<script>
    function disableSubmitButton() {
        document.getElementById("submitBtn").disabled = true;
        document.getElementById("submitBtn").innerHTML = '<i class="fas fa-spinner fa-
spin"></i> Sending...';
    }
</script>
{% endblock %}
users/reset_password.html
{% extends "base.html" %}
{% load static %}

```

```

{% block title %}Reset Password | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/base_style.css' %}">
<link rel="stylesheet" href="{% static 'users/css/reset_password_style.css' %}">
{% endblock %}
{% block content %}
<div class="form-container">
<form method="POST" action="{% url 'reset_password' %}">
    {% csrf_token %}
    <div class="brand-container">
        <a href="{% url 'index' %}">
            
        </a>
        <h1>EduMart</h1>
    </div>
    <h2 class="text-center">Reset Your Password</h2>
    <p class="text-center">Enter a new password to reset your account access.</p>
    {% if messages %}
        <div class="alert alert-danger">
            {% for message in messages %}
                <p>{{ message }}</p>
            {% endfor %}
        </div>
    {% endif %}
    <div class="form-group">
        <div class="form-floating">
            <input type="password" class="form-control" name="new_password" id="new_password" placeholder="New Password" required>
            <label for="new_password">Enter New Password</label>
        </div>
    </div>
    <div class="form-group">
        <div class="form-floating">
            <input type="password" class="form-control" name="confirm_password" id="confirm_password" placeholder="Confirm Password" required>
            <label for="confirm_password">Confirm New Password</label>
        </div>
    </div>
    <button class="btn btn-success w-100" type="submit">Reset Password</button>
    <a href="{% url 'login' %}" class="btn btn-secondary w-100 mt-2">Back to Login</a>
</form>
</div>
{% endblock %}
users/verify_otp.html
{% extends "base.html" %}
{% load static %}

```

```

{% block title %}Verify OTP | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/base_style.css' %}">
<link rel="stylesheet" href="{% static 'users/css/verify_otp_style.css' %}">
{% endblock %}
{% block content %}
<div class="form-container">
<form method="POST" action="{% url 'verify_otp' %}" onsubmit="disableSubmitButton()">
    {% csrf_token %}
    <div class="brand-container">
        <a href="{% url 'index' %}">
            
        </a>
        <h1>EduMart</h1>
    </div>
    <h2 class="text-center">Verify Your OTP</h2>
    <p class="text-center">We have sent a verification code to your email. Please enter it below.</p>
    {% if messages %}
        <div class="alert alert-danger">
            {% for message in messages %}
                <p>{{ message }}</p>
            {% endfor %}
        </div>
    {% endif %}
    <div class="form-group">
        <div class="form-floating">
            <input type="text" name="otp" id="otp" class="form-control" placeholder="Enter OTP" required maxlength="6" pattern="\d{6}">
            <label for="otp">Enter OTP</label>
        </div>
    </div>
    <button id="submitBtn" class="btn btn-primary w-100" type="submit">Verify OTP</button>
    <div class="login-links">
        <a href="{% url 'resend_otp' %}" id="resendOtpLink">Resend OTP</a>
    </div>
</div>
{% endblock %}
{% block script %}
<script>
    function disableSubmitButton() {
        document.getElementById("submitBtn").disabled = true;
        document.getElementById("submitBtn").innerHTML = '<i class="fas fa-spinner fa-spin"></i> Verifying...';
    }
</script>

```

```

        }
        document.addEventListener("DOMContentLoaded", function() {
            let resendLink = document.getElementById("resendOtpLink");
            resendLink.style.pointerEvents = "none";
            resendLink.style.opacity = "0.5";
            setTimeout(function() {
                resendLink.style.pointerEvents = "auto";
                resendLink.style.opacity = "1";
            }, 30000);
        });
    </script>
    {% endblock %}
users/verify_password_otp.html
{% extends "base.html" %}
{% load static %}
{% block title %}Verify OTP | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/base_style.css' %}">
<link rel="stylesheet" href="{% static 'users/css/verify_password_otp_style.css' %}">
{% endblock %}
{% block content %}
<div class="form-container">
    <form method="POST" action="{% url 'verify_password_otp' %}" onsubmit="disableSubmitButton()">
        {% csrf_token %}
        <div class="brand-container">
            <a href="{% url 'index' %}">
                
            </a>
            <h1>EduMart</h1>
        </div>
        <h2 class="text-center">Verify Your OTP</h2>
        <p class="text-center">We have sent a verification code to your email. Please enter it below.</p>
        {% if messages %}
            <div class="alert alert-danger">
                {% for message in messages %}
                    <p>{{ message }}</p>
                {% endfor %}
            </div>
        {% endif %}
        <div class="form-group">
            <div class="form-floating">
                <input type="text" name="otp" id="otp" class="form-control text-center" placeholder="Enter OTP" required maxlength="6" pattern="\d{6}">
                <label for="otp">Enter OTP</label>
            </div>
        </div>
    </form>
</div>

```

```

        </div>
    </div>
    <button id="submitBtn" class="btn btn-primary w-100" type="submit">Verify
OTP</button>
    <div class="login-links">
        <a href="{% url 'resend_otp' %}" id="resendOtpLink">Resend OTP</a>
    </div>
</form>
</div>
{% endblock %}
{% block script %}
<script>
    function disableSubmitButton() {
        document.getElementById("submitBtn").disabled = true;
        document.getElementById("submitBtn").innerHTML = '<i class="fas fa-spinner fa-spin"></i> Verifying...';
    }
    document.addEventListener("DOMContentLoaded", function() {
        let resendLink = document.getElementById("resendOtpLink");
        resendLink.style.pointerEvents = "none";
        resendLink.style.opacity = "0.5";
        setTimeout(function() {
            resendLink.style.pointerEvents = "auto";
            resendLink.style.opacity = "1";
        }, 30000);
    });
</script>
{% endblock %}
users/edit_profile.html
{% extends "base.html" %}
{% load static %}
{% block title %}Edit Profile | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'users/css/register_style.css' %}">
{% endblock %}
{% block content %}
<div class="form-container">
    <form method="POST" action="{% url 'edit_profile' %}" enctype="multipart/form-data">
        {% csrf_token %}
        <div class="brand-container">
            <a href="{% url 'index' %}" class="logo-button">
                
            </a>
            <h1>EduMart</h1>
        </div>
        <h2 class="text-center">Edit Your Profile</h2>

```

```

{%- if messages %}
<div class="alert alert-success">
    {% for message in messages %}
        <p>{{ message }}</p>
    {% endfor %}
</div>
{%- endif %}
<div class="form-group">
    <div class="form-floating mb-3">
        <input type="email" class="form-control" value="{{ user.email }}" disabled>
        <label>Email (Cannot be changed)</label>
    </div>
</div>
<div class="form-group">
    <div class="form-floating mb-3">
        <input type="text" class="form-control" name="username" id="username"
placeholder="Username" value="{{ user.username }}" required>
        <label for="username">Username</label>
    </div>
</div>
<div class="form-group mb-3">
    <div class="form-floating">
        <textarea class="form-control" name="bio" id="bio" placeholder="Bio">{{ user.bio }}

```

**services****admin.py**

```

from django.contrib import admin
from .models import Service
class ServiceAdmin(admin.ModelAdmin):
    list_display = ('title', 'provider', 'category', 'price', 'available', 'created_at')
    search_fields = ('title', 'provider__username', 'category')
    list_filter = ('category', 'available')

```

```

admin.site.register(Service, ServiceAdmin)

apps.py
from django.apps import AppConfig
class ServicesConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'services'

models.py
from django.db import models
from users.models import CustomUser
class Service(models.Model):
    CATEGORY_CHOICES = [('Tutoring', 'Tutoring'), ('Graphic Design', 'Graphic Design'),
    ('App Development', 'App Development'), ('Web Development', 'Web Development'), ('Other',
    'Other')]
    provider = models.ForeignKey(CustomUser, on_delete=models.CASCADE,
related_name='services')
    title = models.CharField(max_length=200)
    description = models.TextField()
    category = models.CharField(max_length=50, choices=CATEGORY_CHOICES)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    image = models.ImageField(upload_to='service_images/', blank=True, null=True)
    available = models.BooleanField(default=True)
    created_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.title

urls.py
from django.urls import path
from .views import service_list, create_service, update_service, delete_service, search_services
urlpatterns = [
    path('search/', search_services, name='search_services'),
    path('', service_list, name="service_list"),
    path('create/', create_service, name="create_service"),
    path('<int:service_id>/update/', update_service, name="update_service"),
    path('<int:service_id>/delete/', delete_service, name="delete_service"),
]
views.py
from django.shortcuts import render, get_object_or_404, redirect
from django.contrib.auth.decorators import login_required
from django.db.models import Q
from django.contrib import messages
from .models import Service
def search_services(request):
    query = request.GET.get('q', "").strip()
    results = []
    if query:
        results = Service.objects.filter(
            Q(title__icontains=query) |

```

```

        Q(category__icontains=query) |
        Q(provider__username__icontains=query)
    ).distinct()
return render(request, 'services/service_list.html', {'services': results, 'query': query})
def service_list(request):
    services = Service.objects.filter(available=True)
    return render(request, "services/service_list.html", {"services": services})
@login_required
def create_service(request):
    category_choices = Service.CATEGORY_CHOICES
    if request.method == "POST":
        title = request.POST.get("title", "").strip()
        description = request.POST.get("description", "").strip()
        category = request.POST.get("category", "").strip()
        other_category = request.POST.get("other_category", "").strip()
        price = request.POST.get("price", "").strip()
        image = request.FILES.get("image")
        available = request.POST.get("available") == "on"
        if category == "Other" and other_category:
            category = other_category.strip()
        if not title or not description or not category or not price:
            messages.error(request, "All fields are required.")
            return render(request, "services/service_form.html", {"category_choices": category_choices})
        Service.objects.create(
            provider=request.user,
            title=title,
            description=description,
            category=category,
            price=price,
            image=image,
            available=available,
        )
        messages.success(request, "Service created successfully!")
        return redirect("profile")
    return render(request, "services/service_form.html", {"category_choices": category_choices})
@login_required
def update_service(request, service_id):
    service = get_object_or_404(Service, id=service_id, provider=request.user)
    category_choices = Service.CATEGORY_CHOICES
    if request.method == "POST":
        title = request.POST.get("title", "").strip()
        description = request.POST.get("description", "").strip()
        category = request.POST.get("category", "").strip()
        other_category = request.POST.get("other_category", "").strip()
        price = request.POST.get("price", "").strip()

```

```

image = request.FILES.get("image")
available = request.POST.get("available") == "on"
if category == "Other" and other_category:
    category = other_category.strip()
if not title or not description or not category or not price:
    messages.error(request, "All fields are required.")
    return render(request, "services/service_form.html", {"service": service,
"category_choices": category_choices})
service.title = title
service.description = description
service.category = category
service.price = price
service.available = available
if image:
    service.image = image
service.save()
messages.success(request, "Service updated successfully!")
return redirect("profile")
return render(request, "services/service_form.html", {"service": service, "category_choices": category_choices})
@login_required
def delete_service(request, service_id):
    service = get_object_or_404(Service, id=service_id, provider=request.user)
    if request.method == "POST":
        service.delete()
        messages.success(request, "Service deleted successfully!")
        return redirect("profile")
    return render(request, "services/service_confirm_delete.html", {"service": service})

```

### static

#### **css/service\_form\_style.css**

```

body {background: linear-gradient(to right, #f8f9fa, #e3f2fd); font-family: 'Poppins', sans-serif;
min-height: 100vh; display: flex; justify-content: center; align-items: center; padding: 50px
15px;}
.container {max-width: 650px; background: #ffffff; padding: 40px; border-radius: 10px; box-
shadow: 0 4px 8px rgba(0, 0, 0, 0.1); margin-top: 20px; margin-bottom: 20px;}
.form-title {display: flex; align-items: center; justify-content: center; gap: 10px; font-weight:
bold; font-size: 1.75rem; color: #007bff; text-align: center; margin-bottom: 20px;}
.form-title img {width: 2rem; height: 2rem;}
.form-label {font-weight: 500; color: #333;}
.form-control {border-radius: 5px; padding: 12px; border: 1px solid #ced4da; transition: all 0.3s
ease-in-out;}
.form-control:focus {border-color: #007bff; box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);}
select.form-control {cursor: pointer;}
img {border-radius: 5px; border: 1px solid #ddd; max-width: 100%; height: auto;}
.form-check-input {accent-color: #007bff;}

```

```
.btn {font-weight: 500; padding: 10px 15px; border-radius: 5px; transition: all 0.3s ease-in-out;}
.btn-primary {background-color: #007bff; border: none;}
.btn-primary:hover {background-color: #0056b3;}
.btn-dark {background-color: #000; border: none; color: white;}
.btn-dark:hover {background-color: #333;}
@media (max-width: 768px) {.container {width: 90%; padding: 25px;}}
css/service_list_style.css
.main-content {width: 100%; max-width: 1400px; margin: 0 auto; display: flex; flex-direction: column; gap: 40px; padding-bottom: 60px; margin-top: 200px;}
.section-title {font-size: 1.8rem; font-weight: bold; color: #004085; border-bottom: 3px solid #004085; padding-bottom: 8px; margin-bottom: 20px; text-align: left; padding-top: 40px;}
.service-list {display: flex; flex-wrap: wrap; gap: 20px; justify-content: center;}
.service-card {flex: 1 1 calc(30% - 20px); background: #ffffff; border-radius: 12px; box-shadow: 0 4px 10px rgba(0, 0, 0, 0.08); padding: 18px; transition: all 0.3s ease-in-out; border: 1px solid #e0e0e0; text-align: left; position: relative; overflow: hidden;}
.service-card:hover {transform: translateY(-5px); box-shadow: 0 6px 15px rgba(0, 0, 0, 0.12);}
.service-image img {width: 100%; height: 180px; object-fit: cover; border-radius: 10px; border: 1px solid #ddd;}
.service-title {font-size: 1.4rem; font-weight: bold; color: #333; margin-top: 12px;}
.service-category, .service-description {font-size: 1rem; color: #555; margin: 4px 0;}
.service-price {font-size: 1.3rem; font-weight: bold; color: #28a745; margin-top: 8px; padding-left: 10px; border-left: 4px solid #28a745;}
.service-provider {font-size: 1rem; color: #007bff; font-weight: bold;}
.service-availability {display: inline-block; padding: 6px 12px; font-size: 0.9rem; font-weight: bold; border-radius: 6px; color: white; margin-top: 8px;}
.available {background: #28a745;}
.not-available {background: #dc3545;}
.service-buttons {display: flex; justify-content: space-between; margin-top: 15px;}
.service-buttons .btn {flex: 1; margin: 5px; border-radius: 6px; font-size: 0.9rem; padding: 10px; transition: all 0.3s ease-in-out;}
.service-buttons .btn-primary {background: #004085; color: #fff;}
.service-buttons .btn-primary:hover {background: #002752;}
.service-buttons .btn-success {background: #28a745; color: #fff;}
.service-buttons .btn-success:hover {background: #1d662d;}
@media (max-width: 1200px) {.service-card {flex: 1 1 calc(45% - 20px);}}
@media (max-width: 768px) {.service-card {flex: 1 1 100%;}.service-buttons {flex-direction: column;}.service-buttons .btn {width: 100%;}}
js/service_form_style.css
function toggleOtherCategory() {
    var categorySelect = document.getElementById("category");
    var otherCategoryDiv = document.getElementById("other-category-div");
    var otherCategoryInput = document.getElementById("other_category");

    if (categorySelect.value === "Other") {
        otherCategoryDiv.style.display = "block";
        otherCategoryInput.required = true;
    }
}
```

```

    } else {
        otherCategoryDiv.style.display = "none";
        otherCategoryInput.required = false;
    }
}
document.addEventListener("DOMContentLoaded", toggleOtherCategory);

```

**templates**

**services/service\_list.html**

```

{% extends "home_base.html" %}
{% load static %}
{% block title %}Available Services | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/service_list_style.css' %}">
{% endblock %}
{% block content %}
<div class="container mt-5">
<h2 class="section-title">
    {% if query %}
        Search Results for "{{ query }}"
    {% else %}
        Available Services
    {% endif %}
</h2>
    {% if services %}
        <div class="service-list">
            {% for service in services %}
                <div class="service-card">
                    <div class="service-image">
                        {% if service.image %}
                            
                        {% else %}
                            
                        {% endif %}
                    </div>
                    <div class="service-info">
                        <h3 class="service-title">{{ service.title }}</h3>
                        <p class="service-category"><strong>Category:</strong> {{ service.category
}}</p>
                        <p class="service-description"><strong>Description:</strong> {{
service.description|truncatewords:20 }}</p>
                        <p class="service-price"><strong>Price:</strong> ₹{{ service.price }}</p>
                        <p class="service-provider"><strong>Provider:</strong> {{
service.provider.username }}</p>
                        <span class="service-availability" {% if service.available %}available
{% else
%}not-available{% endif %}>

```

```

        {% if service.available %}Available{% else %}Not Available{% endif %}
    </span>
    <div class="service-buttons">
        {% if service.available and request.user.is_authenticated and not
request.user.is_service_provider %}
            <a href="{% url 'create_booking' service.id %}" class="btn btn-success">Book
Now</a>
        {% endif %}
            <a href="{% url 'provider_profile' service.provider.id %}" class="btn btn-primary">View Provider Profile</a>
        </div>
    </div>
    {% endfor %}
</div>
{% else %}
<p class="no-services">
    {% if query %}
        No results found for "{{ query }}".
    {% else %}
        No services available at the moment.
    {% endif %}
</p>
{% endif %}
</div>
<br><br>
{% endblock %}
services/service_form.html
{% extends "home_base.html" %}
{% load static %}
{% block title %}{{ service.id|default:"Create" }} Service | EduMart{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/service_form_style.css' %}">
{% endblock %}
{% block content %}
<div class="container mt-5">
    <h2 class="form-title d-flex align-items-center">
        
        {% if service.id %}Update{% else %}Create{% endif %} Service
    </h2>
    <form method="POST" enctype="multipart/form-data" class="shadow p-4 bg-white">
        {% csrf_token %}
        <!-- Title -->
        <div class="mb-3">
            <label for="title" class="form-label">Service Title</label>

```

```

<input type="text" id="title" name="title" class="form-control" required value="{{ service.title|default:" " }}>
</div>
<!-- Description -->
<div class="mb-3">
    <label for="description" class="form-label">Description</label>
    <textarea id="description" name="description" class="form-control" rows="4" required>{{ service.description|default:" " }}</textarea>
</div>
<!-- Category -->
<div class="mb-3">
    <label for="category" class="form-label">Category</label>
    <select id="category" name="category" class="form-control" required onchange="toggleOtherCategory()">
        {% with category_keys=category_choices.keys %}
            {% for key, value in category_choices %}
                <option value="{{ key }}" {% if service.category == key %}selected{% endif %}>{{ value }}</option>
            {% endfor %}
            {% endwith %}
        </select>
    </div>
    <!-- Other Category (Hidden by Default, Visible if "Other" is Selected) -->
    <div class="mb-3" id="other-category-div" {% if service.category in category_choices.keys %}style="display: none;"{% endif %}>
        <label for="other_category" class="form-label">Enter Custom Category</label>
        <input type="text" id="other_category" name="other_category" class="form-control" value="{{% if service.category not in category_choices.keys %}} {{ service.category }}{{% endif %}}">
    </div>
    <!-- Price -->
    <div class="mb-3">
        <label for="price" class="form-label">Price (INR ₹)</label>
        <input type="number" id="price" name="price" class="form-control" required min="0" value="{{ service.price|default:" 0 }}"/>
    </div>
    <!-- Image Upload -->
    <div class="mb-3">
        <label class="form-label">Service Image</label>
        {% if service.image %}
            <div>
                
            </div>
        {% endif %}
        <input type="file" name="image" class="form-control">
    </div>

```

```

</div>
<!-- Availability -->
<div class="form-check mb-3">
    <input type="checkbox" id="available" name="available" class="form-check-input" {%
if service.available %}checked{%
endif %}>
    <label for="available" class="form-check-label">Available</label>
</div>
<!-- Submit and Cancel Buttons -->
<button type="submit" class="btn btn-primary">{%
if service.id %}Update{%
else %}Create{%
endif %} Service</button>
<a href="{%
url 'profile' %}" class="btn btn-dark">Cancel</a>
</form>
</div>
{%
endblock %}
{%
block script %}
<script src="{%
static 'js/service_form_script.js' %}"></script>
{%
endblock %}
services/service_confirm_delete.html
{%
extends "base.html" %}
{%
block title %}Delete Service | EduMart{%
endblock %}
{%
block content %}
<div class="container mt-5 text-center">
    <h2 class="text-danger">Are you sure you want to delete this service?</h2>
    <p class="lead">You are about to delete <strong>{{ service.title }}</strong>.</p>
    <form method="POST">
        {%
        csrf_token %}
        <button type="submit" class="btn btn-danger">Delete</button>
        <a href="{%
url 'service_list' %}" class="btn btn-secondary">Cancel</a>
    </form>
</div>
{%
endblock %}

```

**bookings****admin.py**

```

from django.contrib import admin
from .models import Booking
class BookingAdmin(admin.ModelAdmin):
    list_display = ('client', 'service', 'date', 'time', 'status', 'created_at')
    search_fields = ('client__username', 'service__title')
    list_filter = ('status', 'date')
admin.site.register(Booking, BookingAdmin)

```

**apps.py**

```

from django.apps import AppConfig
class BookingsConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'bookings'

```

**models.py**

```

from django.db import models
from users.models import CustomUser
from services.models import Service
class Booking(models.Model):
    STATUS_CHOICES = [
        ('Pending', 'Pending'),
        ('Approved', 'Approved'),
        ('Confirmed', 'Confirmed'),
        ('Cancelled', 'Cancelled'),
        ('Completed', 'Completed'),
    ]
    client = models.ForeignKey(CustomUser, on_delete=models.CASCADE,
related_name='bookings')
    service = models.ForeignKey(Service, on_delete=models.CASCADE,
related_name='bookings')
    date = models.DateField()
    time = models.TimeField()
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='Pending')
    created_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return f'{self.client.username} - {self.service.title} ({self.date} {self.time})"
models.py
from django.urls import path
from .views import booking_list, create_booking, cancel_booking, accept_booking,
reject_booking

urlpatterns = [
    path("", booking_list, name="booking_list"),
    path('create/<int:service_id>', create_booking, name="create_booking"),
    path('<int:booking_id>/cancel', cancel_booking, name="cancel_booking"),
    path('<int:booking_id>/accept', accept_booking, name="accept_booking"),
    path('<int:booking_id>/reject', reject_booking, name="reject_booking"),
]
views.py
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.decorators import login_required
from django.contrib import messages
from services.models import Service
from .models import Booking
from datetime import timedelta
from django.utils import timezone
@login_required
def booking_list(request):
    bookings = Booking.objects.filter(client=request.user).order_by('-created_at')
    return render(request, "bookings/booking_list.html", {"bookings": bookings})
@login_required

```

```

def create_booking(request, service_id):
    service = get_object_or_404(Service, id=service_id)
    if request.method == "POST":
        date, time = request.POST.get("date"), request.POST.get("time")
        if not date or not time:
            messages.error(request, "Please provide both date and time.")
            return render(request, "bookings/booking_form.html", {"service": service})
        try:
            selected_date = timezone.datetime.strptime(date, "%Y-%m-%d").date()
        except ValueError:
            messages.error(request, "Invalid date format.")
            return render(request, "bookings/booking_form.html", {"service": service})
        if selected_date < timezone.localdate() + timedelta(days=1):
            messages.error(request, "Bookings must be made at least one day in advance.")
            return render(request, "bookings/booking_form.html", {"service": service})
        if Booking.objects.filter(client=request.user, service=service,
                                   service_provider=service.provider).exclude(status__in=["Completed", "Cancelled"]).exists():
            messages.error(request, "You have an ongoing booking for this service.")
            return render(request, "bookings/booking_form.html", {"service": service})
        if Booking.objects.filter(service=service, date=selected_date,
                                   time=time).exclude(status="Cancelled").exists():
            messages.error(request, "This slot is already booked.")
            return render(request, "bookings/booking_form.html", {"service": service})
        Booking.objects.create(client=request.user, service=service, date=selected_date, time=time,
                               status="Pending")
        messages.success(request, "Booking request sent successfully.")
        return redirect("profile")
    return render(request, "bookings/booking_form.html", {"service": service})
@login_required
def cancel_booking(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id, client=request.user)
    if request.method == "POST":
        if booking.status in ["Pending", "Approved", "Confirmed"]:
            booking.status = "Cancelled"
            booking.save()
            messages.success(request, "Booking cancelled successfully.")
        else:
            messages.error(request, "This booking cannot be cancelled.")
            return redirect("profile")
    return render(request, "bookings/booking_cancel.html", {"booking": booking})
@login_required
def accept_booking(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id, service_provider=request.user)
    if booking.status == "Pending":
        booking.status = "Approved"
        booking.save()

```

```

        messages.success(request, "Booking approved! Waiting for client payment.")
else:
    messages.error(request, "This booking cannot be approved.")
    return redirect("profile")
@login_required
def reject_booking(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id, service_provider=request.user)
    if request.method == "POST":
        if booking.status == "Pending":
            booking.status = "Cancelled"
            booking.save()
            messages.success(request, "Booking rejected successfully.")
        else:
            messages.error(request, "This booking cannot be rejected.")
            return redirect("profile")
    return render(request, "bookings/booking_reject.html", {"booking": booking, "action": "Reject"})

```

**static****css/booking\_service\_style.css**

```

.booking-container {max-width: 750px; background: #fff; padding: 35px; border-radius: 12px;
box-shadow: 0px 6px 18px rgba(0,0,0,0.15); margin: 130px auto 20px auto; transition: transform
0.3s ease-in-out, box-shadow 0.3s ease-in-out;}
.booking-container:hover {transform: translateY(-5px); box-shadow: 0px 10px 25px
rgba(0,0,0,0.2);}
.service-info-card {display: flex; align-items: center; background: linear-
gradient(135deg,#f8f9fa,#eef1f6); padding: 20px; border-radius: 12px; box-shadow: 0px 4px
12px rgba(0,0,0,0.1); margin-bottom: 25px; transition: all 0.3s ease-in-out;}
.service-img {width: 130px; height: 130px; border-radius: 12px; object-fit: cover; margin-right:
20px; box-shadow: 0px 4px 8px rgba(0,0,0,0.15); transition: transform 0.3s ease-in-out;}
.service-img:hover {transform: scale(1.05);}
.service-details {flex-grow: 1;}
.service-details p {margin: 6px 0; font-size: 17px; font-weight: 500; color: #333;}
.service-title {font-size: 28px; font-weight: 700; color: #0078E8; margin-bottom: 5px; text-
transform: capitalize;}
.service-details p strong {color: #0078E8; font-weight: 600;}
.input-group {position: relative; margin-bottom: 22px;}
.floating-input {width: 100%; padding: 16px 14px; font-size: 18px; border: 1px solid #ced4da;
border-radius: 8px; background: #fff; transition: all 0.3s ease-in-out; outline: none;}
.floating-label {position: absolute; top: 50%; left: 16px; transform: translateY(-50%); font-size:
16px; color: #999; transition: all 0.3s ease-in-out; pointer-events: none;}
.floating-input:focus + .floating-label, .floating-input:not(:placeholder-shown) + .floating-label
{top: 10px; font-size: 13px; color: #0078E8;}
.floating-input:focus {border-color: #0078E8; box-shadow: 0px 0px 12px rgba(0,120,232,0.3);
transform: scale(1.02);}
.booking-form .btn {padding: 14px 22px; font-size: 17px; font-weight: 600; border-radius: 6px;
transition: all 0.3s ease-in-out; text-transform: uppercase;}

```

```
.booking-form .btn-success {background: linear-gradient(135deg,#28a745,#218838); border: none;}
.booking-form .btn-success:hover {background: linear-gradient(135deg,#218838,#1c6b30); transform: scale(1.07);}
.booking-form .btn-secondary {background: linear-gradient(135deg,#6c757d,#545b62); border: none;}
.booking-form .btn-secondary:hover {background: linear-gradient(135deg,#545b62,#3e444a); transform: scale(1.07);}
@media (max-width: 576px) {
    .booking-container {width: 90%;}
    .service-info-card {flex-direction: column; text-align: center; padding: 15px;}
    .service-img {margin-bottom: 12px;}
    .service-title {font-size: 26px;}
    .booking-form .btn {width: 100%; margin-bottom: 12px;}
}
```

### templates

#### **bookings/booking\_form.html**

```
{% extends "home_base.html" %}
{% load static %}

{% block title %}Booking Service | {{ service.title }}{% endblock %}
{% block style %}<link rel="stylesheet" href="{% static 'css/booking_service_style.css' %}">{% endblock %}
{% block content %}

<div class="container booking-container">
    <div class="service-info-card">
        {% if service.image %}{% endif %}
        <div class="service-details">
            <h2 class="service-title">{{ service.title }}</h2>
            <p><strong>Provider:</strong> {{ service.provider.username }}</p>
            <p><strong>Price:</strong> ₹{{ service.price }}</p>
            <p><strong>Description:</strong> {{ service.description }}</p>
        </div>
    </div>
    {% if messages %}{% for message in messages %}<div class="alert alert-{{ message.tags }}">{{ message }}</div>{% endfor %}{% endif %}
    <form method="POST" class="booking-form">
        {% csrf_token %}
        <div class="input-group"><input type="date" name="date" id="date" class="form-control floating-input" required><label for="date" class="floating-label">Select Date</label></div>
        <div class="input-group"><input type="time" name="time" id="time" class="form-control floating-input" required><label for="time" class="floating-label">Select Time</label></div>
        <button type="submit" class="btn btn-success">Confirm Booking</button>
        <a href="{% url 'profile' %}" class="btn btn-secondary">Cancel</a>
    </form>
</div>
```

```

{%
  endblock %}
{%
  block script %}
<script>
document.addEventListener("DOMContentLoaded", function () {
  let today = new Date(); today.setDate(today.getDate() + 1);
  let minDate = today.toISOString().split("T")[0];
  document.getElementById("date").setAttribute("min", minDate);
});
</script>
{%
  endblock %}
bookings/booking_reject.html
{%
  extends "home_base.html" %
}
{%
  block content %
}
<div class="container mt-4">
  <div class="card shadow-sm p-4 bg-light">
    <h2 class="mb-3 text-danger">Reject Booking</h2>
    <p class="mb-4">Are you sure you want to reject booking for <strong>"{{ booking.service.title }}</strong> on <strong>"{{ booking.date }}</strong> at <strong>"{{ booking.time }}</strong>?</p>
    <form method="POST">
      {% csrf_token %}
      <button type="submit" class="btn btn-danger">Confirm Rejection</button>
      <a href="{% url 'profile' %}" class="btn btn-secondary">Go Back</a>
    </form>
  </div>
</div>
{%
  endblock %}
bookings/booking_cancel.html
{%
  extends "home_base.html" %
}
{%
  block content %
}
<div class="container mt-4">
  <div class="card shadow-sm p-4 bg-light">
    <h2 class="mb-3 text-danger">Cancel Booking</h2>
    <p class="mb-4">Are you sure you want to cancel your booking for <strong>"{{ booking.service.title }}</strong> on <strong>"{{ booking.date }}</strong> at <strong>"{{ booking.time }}</strong>?</p>
    <form method="POST">
      {% csrf_token %}
      <button type="submit" class="btn btn-danger">Confirm Cancellation</button>
      <a href="{% url 'profile' %}" class="btn btn-secondary">Go Back</a>
    </form>
  </div>
</div>
{%
  endblock %}

```

**Payments****admin.py**

```

from django.contrib import admin
from .models import Payment
class PaymentAdmin(admin.ModelAdmin):
    list_display = ('booking', 'payment_id', 'amount', 'status', 'created_at')
    search_fields = ('payment_id', 'booking__client__username')
    list_filter = ('status',)
admin.site.register(Payment, PaymentAdmin)

```

**apps.py**

```

from django.apps import AppConfig
class PaymentsConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'payments'

```

**models.py**

```

from django.db import models
from bookings.models import Booking
class Payment(models.Model):
    STATUS_CHOICES = [
        ('Pending', 'Pending'),
        ('Completed', 'Completed'),
        ('Failed', 'Failed'),
    ]
    booking = models.OneToOneField(Booking, on_delete=models.CASCADE,
related_name='payment')
    payment_id = models.CharField(max_length=100, blank=True, null=True)
    amount = models.DecimalField(max_digits=10, decimal_places=2)
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='Pending')
    created_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return f"Payment {self.payment_id} - {self.status}"

```

**urls.py**

```

from django.urls import path
from .views import make_payment, initiate_payment, payment_success
urlpatterns = [
    path('pay/<int:booking_id>', make_payment, name="make_payment"),
    path('initiate/<int:booking_id>', initiate_payment, name="initiate_payment"),
    path('success/', payment_success, name="payment_success"),
]

```

**views.py**

```

import razorpay
import logging
from django.conf import settings
from django.shortcuts import render, redirect, get_object_or_404
from django.http import JsonResponse
from django.contrib import messages
from django.urls import reverse
from bookings.models import Booking

```

```

from .models import Payment
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)
razorpay_client = razorpay.Client(auth=(settings.RAZORPAY_API_KEY,
settings.RAZORPAY_API_SECRET))
def make_payment(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id, client=request.user)
    if Payment.objects.filter(booking=booking, status="Completed").exists():
        messages.info(request, "This booking has already been paid for.")
        return redirect("profile")
    return redirect("initiate_payment", booking_id=booking.id)
def initiate_payment(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id, client=request.user)
    try:
        payment = Payment.objects.get(booking=booking)
        if payment.status == "Completed":
            messages.info(request, "This booking has already been paid for.")
            return redirect("profile")
        amount = int(booking.service.price * 100)
    except Payment.DoesNotExist:
        amount = int(booking.service.price * 100)
    try:
        order_data = {"amount": amount, "currency": "INR", "payment_capture": 1}
        order = razorpay_client.order.create(order_data)
        logger.info(f"Created Razorpay order: {order['id']} ")
        payment, created = Payment.objects.update_or_create(
            booking=booking,
            defaults={
                'payment_id': order['id'],
                'amount': booking.service.price,
                'status': "Pending"
            }
        )
        test_cards = {
            "success": {"number": "4111 1111 1111 1111", "cvv": "123", "expiry": "12/25"},
            "failure": {"number": "4242 4242 4242 4242", "cvv": "123", "expiry": "12/25"}
        }
        context = {
            "booking": booking,
            "order_id": order['id'],
            "razorpay_merchant_key": settings.RAZORPAY_API_KEY,
            "amount": amount,
            "currency": "INR",
            "test_cards": test_cards,
            "callback_url": request.build_absolute_uri(reverse('payment_success'))
        }
    
```

```

    logger.info(f'Rendering payment page for booking {booking_id}')
    return render(request, "payments/payment_page.html", context)
except Exception as e:
    logger.error(f'Payment initiation failed: {str(e)}')
    messages.error(request, "Failed to initiate payment. Please try again.")
    return redirect("profile")
def payment_success(request):
    if request.method == "POST":
        try:
            params_dict = {
                'razorpay_payment_id': request.POST.get('razorpay_payment_id'),
                'razorpay_order_id': request.POST.get('razorpay_order_id'),
                'razorpay_signature': request.POST.get('razorpay_signature')
            }
            logger.info(f'Received payment callback: {params_dict}')
            try:
                razorpay_client.utility.verify_payment_signature(params_dict)
                payment = Payment.objects.get(payment_id=params_dict['razorpay_order_id'])
                payment.status = "Completed"
                payment.save()
                booking = payment.booking
                booking.status = "Confirmed"
                booking.save()
                messages.success(request, "Payment successful! Your booking is confirmed.")
                return redirect('profile')
            except Exception as e:
                logger.error(f'Payment verification failed: {str(e)}')
                messages.error(request, "Payment verification failed. Please contact support.")
                return redirect('profile')
        except Exception as e:
            logger.error(f'Payment processing failed: {str(e)}')
            messages.error(request, "Payment processing failed. Please try again.")
            return redirect('profile')
    return redirect('profile')

```

**static****css/payment\_style.css**

```

body {background-color:#f4f6f9;font-family:'Poppins',sans-serif;}
.container {max-width:1200px;padding:12px;}
.card {border-radius:8px;box-shadow:0px 4px 12px rgba(0,0,0,0.12);overflow:hidden;background-color:#ffffff;padding:8px;border:none;}
.card-header {font-size:1.4rem;font-weight:700;padding:12px;text-align:center;background-color:#0078E8;color:#ffffff;border-bottom:3px solid #005bbf;}
.alert {border-radius:6px;padding:10px;font-size:0.9rem;margin-bottom:12px;box-shadow:0px 2px 6px rgba(0,0,0,0.08);}
.card-header.bg-success,.card-header.bg-danger {font-size:1rem;text-align:center;padding:6px;}
.card-body {padding:10px;}

```

```
.booking-details h4 {text-align:center;margin-bottom:10px;font-size:1.3rem;font-weight:700;color:#0078E8;}
.table {border-radius:6px;overflow:hidden;width:100%;}
.table th {width:40%;text-align:left;font-weight:600;color:#0078E8;padding:10px;background-color:#f1f1f1;border-bottom:2px solid #ddd;}
.table td {width:60%;text-align:right;font-size:1rem;font-weight:bold;color:#0078E8;padding:10px;background-color:#ffffff;border-bottom:2px solid #ddd;}
.text-center {margin-top:12px;display:flex;justify-content:space-between;gap:12px;width:100%;}
.btn-lg {font-size:1rem;padding:12px;border-radius:8px;font-weight:600;transition:all 0.3s ease-in-out;width:48%;max-width:450px;text-align:center;display:flex;align-items:center;justify-content:center;box-shadow:0px 4px 8px rgba(0,0,0,0.15);}
.btn-primary {background-color:#0078E8;border:2px solid #0078E8;color:white;transition:all 0.3s ease-in-out;}
.btn-primary:hover {background-color:#005bbf;}
.btn-secondary {background-color:#dc3545;border:2px solid #dc3545;color:white;transition:all 0.3s ease-in-out;}
.btn-secondary:hover {background-color:#bb2d3b;}
@media (max-width:992px) {
.container {max-width:96%;}
.table th,.table td {text-align:left;display:block;width:100%;}
.text-center {flex-direction:column;gap:8px;}
.btn-lg {width:100%;}
}

```

## templates

### **payments/payment\_page.html**

```
{% extends 'home_base.html' %}
{% load static %}
{% block title %} Payment | {{ booking.service.title }} {% endblock %}
{% block style %} <link rel="stylesheet" href="{% static 'css/payment_style.css' %}"> {% endblock %}
{% block content %}
<br>
<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header bg-primary text-white">
                    <h3 class="mb-0">Secure Payment</h3>
                </div>
                <div class="card-body">
                    <!-- Test Mode Instructions -->
                    <div class="alert alert-info">
                        <h5 class="alert-heading">Test Mode Instructions</h5>
                        <p>Use these test card details:</p>

```

```

<div class="row">
    <div class="col-md-6">
        <div class="card mb-3">
            <div class="card-header bg-success text-white">
                Success Card
            </div>
            <div class="card-body">
                <p><strong>Card Number:</strong> {{ test_cards.success.number }}</p>
                <p><strong>CVV:</strong> {{ test_cards.success.cvv }}</p>
                <p><strong>Expiry:</strong> {{ test_cards.success.expiry }}</p>
            </div>
        </div>
    </div>
    <div class="col-md-6">
        <div class="card mb-3">
            <div class="card-header bg-danger text-white">
                Failure Card
            </div>
            <div class="card-body">
                <p><strong>Card Number:</strong> {{ test_cards.failure.number }}</p>
                <p><strong>CVV:</strong> {{ test_cards.failure.cvv }}</p>
                <p><strong>Expiry:</strong> {{ test_cards.failure.expiry }}</p>
            </div>
        </div>
    </div>
</div>
<!-- Booking Details -->
<div class="booking-details mb-4">
    <h4>Booking Details</h4>
    <div class="table-responsive">
        <table class="table">
            <tr>
                <th>Service:</th>
                <td>{{ booking.service.title }}</td>
            </tr>
            <tr>
                <th>Provider:</th>
                <td>{{ booking.service.provider.username }}</td>
            </tr>
            <tr>
                <th>Amount:</th>
                <td>₹{{ widthratio amount 100 1 }}.00</td>
            </tr>
        </table>
    </div>
</div>

```

```

        </table>
    </div>
</div>
<!-- Payment Buttons -->
<div class="text-center">
    <button id="rzp-button" class="btn btn-primary btn-lg">
        <i class="fas fa-lock mr-2"></i>Pay Securely
    </button>
    <a href="{% url 'profile' %}" class="btn btn-secondary btn-lg ml-2">
        <i class="fas fa-times mr-2"></i>Cancel
    </a>
</div>
</div>
</div>
</div>
<br>
{% endblock %}
{% block script %}
<!-- Razorpay Integration -->
<script src="https://checkout.razorpay.com/v1/checkout.js"></script>
<script>
    var options = {
        "key": "{{ razorpay_merchant_key }}",
        "amount": "{{ amount }}",
        "currency": "{{ currency }}",
        "name": "EduMart",
        "description": "Payment for {{ booking.service.title }}",
        "order_id": "{{ order_id }}",
        "handler": function (response) {
            // Create form for payment verification
            var form = document.createElement('form');
            form.method = 'POST';
            form.action = "{{ callback_url }}";
            // Add CSRF token
            var csrf = document.createElement('input');
            csrf.name = "csrfmiddlewaretoken";
            csrf.value = "{{ csrf_token }}";
            csrf.type = 'hidden';
            form.appendChild(csrf);
            // Add payment response fields
            var fields = ['razorpay_payment_id', 'razorpay_order_id', 'razorpay_signature'];
            fields.forEach(field => {
                if(response[field]){
                    var input = document.createElement('input');

```

```

        input.name = field;
        input.value = response[field];
        input.type = 'hidden';
        form.appendChild(input);
    }
});

// Submit form
document.body.appendChild(form);
form.submit();
},
"prefill": {
    "name": "{{ request.user.get_full_name }}",
    "email": "{{ request.user.email }}",
    "contact": "{{ request.user.phone_number|default:" }}"
},
"theme": {
    "color": "#3399cc"
},
"modal": {
    "ondismiss": function(){
        window.location.href = "{% url 'profile' %}";
    }
};
// Initialize Razorpay
var rzp = new Razorpay(options);
// Handle payment button click
document.getElementById('rzp-button').onclick = function(e){
    rzp.open();
    e.preventDefault();
}
</script>
{% endblock %}

```

**chat****admin.py**

```

from django.contrib import admin
from .models import ChatMessage
@admin.register(ChatMessage)
class ChatMessageAdmin(admin.ModelAdmin):
    list_display = ('id', 'sender', 'receiver', 'message', 'timestamp')
    search_fields = ('sender__username', 'receiver__username', 'message')
    list_filter = ('timestamp',)

```

**apps.py**

```

from django.apps import AppConfig
class ChatConfig(AppConfig):
    default_auto_field = "django.db.models.BigAutoField"

```

```

name = "chat"
def ready(self):
    import chat.signals
consumers.py
import json
from channels.generic.websocket import AsyncWebSocketConsumer
from users.models import CustomUser as User
from .models import ChatMessage
from bookings.models import Booking
from django.core.exceptions import ObjectDoesNotExist

class ChatConsumer(AsyncWebSocketConsumer):
    async def connect(self):
        self.room_name = self.scope['url_route']['kwargs']['booking_id']
        self.room_group_name = f'chat_{self.room_name}'
        await self.channel_layer.group_add(self.room_group_name, self.channel_name)
        await self.accept()

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.room_group_name, self.channel_name)

    async def receive(self, text_data):
        data = json.loads(text_data)
        message = data.get('message', "").strip()
        receiver_id = data.get('receiver_id')
        sender = self.scope["user"]
        if not message:
            return
        try:
            booking = await Booking.objects.aget(id=int(self.room_name))
        except ObjectDoesNotExist:
            return
        try:
            receiver = await User.objects.aget(id=receiver_id)
        except ObjectDoesNotExist:
            return
        chat_message = await ChatMessage.objects.acreate(
            sender=sender,
            receiver=receiver,
            message=message,
            booking=booking
        )
        await self.send_chat_message(chat_message, sender.username, receiver.username)

    async def send_chat_message(self, chat_message, sender_username, receiver_username):
        await self.channel_layer.group_send(

```

```

        self.room_group_name,
    {
        'type': 'chat_message',
        'message': chat_message.message,
        'sender': sender_username,
        'receiver': receiver_username,
        'timestamp': chat_message.timestamp.strftime("%Y-%m-%d %H:%M:%S")
    }
)

async def chat_message(self, event):
    await self.send(text_data=json.dumps({
        'message': event['message'],
        'sender': event['sender'],
        'receiver': event['receiver'],
        'timestamp': event['timestamp']
    }))

```

**models.py**

```

from django.db import models
from users.models import CustomUser as User
from bookings.models import Booking
class ChatMessage(models.Model):
    sender = models.ForeignKey(User, on_delete=models.CASCADE,
    related_name="sent_messages")
    receiver = models.ForeignKey(User, on_delete=models.CASCADE,
    related_name="received_messages")
    booking = models.ForeignKey(Booking, on_delete=models.CASCADE)
    message = models.TextField()
    timestamp = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return f'{self.sender.username} -> {self.receiver.username}: {self.message[:20]}'

```

**routing.py**

```

from django.urls import re_path
from chat.consumers import ChatConsumer
websocket_urlpatterns = [
    re_path(r'^ws/chat/(?P<booking_id>\d+)/$', ChatConsumer.as_asgi()),
]

```

**signals.py**

```

from django.db.models.signals import post_save
from django.dispatch import receiver
from bookings.models import Booking
from chat.models import ChatMessage
@receiver(post_save, sender=Booking)
def delete_chat_on_completion(sender, instance, **kwargs):
    if instance.status == "Completed":
        ChatMessage.objects.filter(

```

```

        sender=instance.client, receiver=instance.service.provider
    ).delete()
    ChatMessage.objects.filter(
        sender=instance.service.provider, receiver=instance.client
    ).delete()

urls.py
from django.urls import path
from .views import chat_room, fetch_messages
urlpatterns = [
    path('<int:booking_id>', chat_room, name='chat_room'),
    path('messages/<int:booking_id>', fetch_messages, name='fetch_messages'),
]

views.py
from django.shortcuts import render, get_object_or_404
from django.http import JsonResponse, HttpResponseRedirect
from django.contrib.auth.decorators import login_required
from django.db.models import Q
from users.models import CustomUser as User
from .models import ChatMessage
from bookings.models import Booking
@login_required
def chat_room(request, booking_id):
    booking = get_object_or_404(
        Booking.objects.select_related("service_provider", "client"),
        id=booking_id,
        status="Confirmed"
    )
    provider = booking.service.provider
    client = booking.client
    if request.user not in [client, provider]:
        return HttpResponseRedirect("Unauthenticated access")
    receiver = client if request.user == provider else provider
    messages = ChatMessage.objects.filter(
        booking=booking
    ).order_by("timestamp")
    return render(request, "chat/chat_room.html", {
        "booking": booking,
        "provider": provider,
        "client": client,
        "messages": messages,
        "receiver": receiver,
    })
@login_required
def fetch_messages(request, booking_id):
    booking = get_object_or_404(Booking, id=booking_id, status="Confirmed")
    provider = booking.service.provider

```

```

client = booking.client
if request.user not in [client, provider]:
    return JsonResponse({"error": "Unauthorized"}, status=403)
messages = ChatMessage.objects.filter(
    booking=booking
).order_by("timestamp")
return JsonResponse({
    "messages": [
        {
            "sender": msg.sender.username,
            "receiver": msg.receiver.username,
            "message": msg.message,
            "timestamp": msg.timestamp.strftime("%Y-%m-%d %H:%M:%S")
        } for msg in messages
    ]
})

```

**static****css/chat\_style.css**

```

.chat-page-container {display: flex; justify-content: center; align-items: center; height: 100vh; width: 100%; background: #eef2f7; padding: 20px;}
.chat-container {width: 50vw; max-width: 1200px; background: #ffffff; padding: 20px; border-radius: 15px; box-shadow: 0 5px 20px rgba(0, 0, 0, 0.2); display: flex; flex-direction: column; height: 85vh; border: 2px solid #ddd;}
.chat-header {display: flex; align-items: center; border-bottom: 3px solid #007bff; padding-bottom: 15px; margin-bottom: 15px; background: #f8f9fa; padding: 10px; border-radius: 10px;}
.chat-profile-pic {width: 40px; height: 40px; border-radius: 50%; margin-right: 12px; object-fit: cover; border: 2px solid #007bff;}
.chat-title {font-size: 1.8rem; color: #007bff; font-weight: bold;}
.chat-title span {color: #0056b3;}
#chat-box {flex-grow: 1; overflow-y: auto; padding: 15px; border: 2px solid #ddd; border-radius: 10px; background: #f9f9f9; display: flex; flex-direction: column; gap: 12px; width: 100%; max-height: 65vh;}
.chat-message {max-width: 75%; padding: 12px; border-radius: 12px; word-wrap: break-word; display: flex; flex-direction: column; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); border: 1px solid rgba(0, 0, 0, 0.1);}
.sent {background-color: #007bff; color: #fff; align-self: flex-end; text-align: left; border-top-right-radius: 0; font-weight: 600; padding: 14px; line-height: 1.6; font-size: 1.1rem; box-shadow: 0 3px 8px rgba(0, 123, 255, 0.2);}
.sent .timestamp {color: rgba(255, 255, 255, 0.8); font-weight: 500;}
.received {background-color: #dbe2ef; color: #222; align-self: flex-start; border-top-left-radius: 0; font-weight: 500; padding: 14px; line-height: 1.5; font-size: 1.1rem; box-shadow: 0 3px 8px rgba(0, 0, 0, 0.1);}
.message-text {font-size: 1rem;}
.timestamp {font-size: 0.75rem; color: gray; text-align: right; margin-top: 5px;}
#chat-form {display: flex; margin-top: 15px; background: #f8f9fa; padding: 10px; border-radius: 10px; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);}

```

```

#message-input {flex: 1; padding: 12px; border: 2px solid #ddd; border-radius: 8px; font-size: 1rem;}
#send-button {padding: 12px 18px; border: none; background-color: #007bff; color: white; border-radius: 8px; margin-left: 8px; cursor: pointer; font-size: 1rem; transition: background 0.3s;}
#send-button:hover {background-color: #0056b3;}
js/chat.js
document.addEventListener("DOMContentLoaded", function () {
    const chatBox = document.getElementById("chat-box");
    const messageInput = document.getElementById("message-input");
    const sendButton = document.getElementById("send-button");
    const chatForm = document.getElementById("chat-form");
    const bookingId = chatBox.dataset.bookingId;
    const currentUser = chatBox.dataset.currentUser;
    const receiverId = chatBox.dataset.receiverId;
    let chatSocket;
    function connectWebSocket() {
        const protocol = window.location.protocol === "https:" ? "wss" : "ws";
        chatSocket = new WebSocket(` ${protocol}://${window.location.host}/ws/chat/${bookingId}/`);
        chatSocket.onopen = function () {
            console.log("WebSocket connected.");
        };
        chatSocket.onmessage = function (event) {
            const data = JSON.parse(event.data);
            const isSender = data.sender === currentUser;
            appendMessage(data.message, data.timestamp, isSender);
        };
        chatSocket.onclose = function (event) {
            console.log("WebSocket disconnected.", event);
            setTimeout(connectWebSocket, 3000);
        };
        chatSocket.onerror = function (error) {
            console.error("WebSocket error:", error);
            chatSocket.close();
        };
    }
    function appendMessage(message, timestamp, isSender) {
        const messageDiv = document.createElement("div");
        messageDiv.classList.add("chat-message", isSender ? "sent" : "received");
        messageDiv.innerHTML = `<p class="message-text">${message}</p><span class="timestamp">${timestamp}</span>`;
        chatBox.appendChild(messageDiv);
        chatBox.scrollTop = chatBox.scrollHeight;
    }
    chatForm.addEventListener("submit", function (event) {

```

```

event.preventDefault();
const message = messageInput.value.trim();
if (message !== "" && receiverId) {
    chatSocket.send(JSON.stringify({message: message, receiver_id: receiverId}));
    messageInput.value = "";
} else {
    console.error("Message is empty or receiver ID is missing!");
}
});
chatBox.scrollTop = chatBox.scrollHeight;
connectWebSocket();
});

```

### templates

#### **chat/chat\_room.html**

```

{% extends 'home_base.html' %}
{% load static %}
{% block title %}Chat with {{ receiver.username }}{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/chat_style.css' %}">
{% endblock %}
{% block content %}
<div class="chat-page-container">
    <div class="chat-container">
        <div class="chat-header">
            
            <h2 class="chat-title">Chat with <span>{{ receiver.username }}</span></h2>
        </div>
        <div id="chat-box" data-booking-id="{{ booking.id }}" data-receiver-id="{{ provider.id }}"
            data-current-user="{{ request.user.username }}">
            {% for message in messages %}
                <div class="chat-message" {% if message.sender == request.user %}sent{% else
                %}received{% endif %}>
                    <p class="message-text">{{ message.message }}</p>
                    <span class="timestamp">{{ message.timestamp }}</span>
                </div>
            {% empty %}
            <p class="no-messages">No messages yet. Start the conversation!</p>
            {% endfor %}
        </div>
        <form id="chat-form">
            <input type="text" id="message-input" placeholder="Type your message..." autocomplete="off" required>
            <button type="submit" id="send-button">Send</button>
        </form>
    </div>
</div>

```

```
{% endblock %}
{% block script %}
<script src="{% static 'js/chat.js' %}"></script>
{% endblock %}
```

**reviews****admin.py**

```
from django.contrib import admin
from .models import Review
class ReviewAdmin(admin.ModelAdmin):
    list_display = ('service', 'client', 'rating', 'created_at')
    search_fields = ('service__title', 'client__username')
    list_filter = ('rating',)
admin.site.register(Review, ReviewAdmin)
```

**apps.py**

```
from django.apps import AppConfig
class ReviewsConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'reviews'
```

**models.py**

```
from django.db import models
from users.models import CustomUser
from services.models import Service
class Review(models.Model):
    service = models.ForeignKey(Service, on_delete=models.CASCADE,
related_name='reviews')
    client = models.ForeignKey(CustomUser, on_delete=models.CASCADE,
related_name='reviews')
    rating = models.IntegerField(choices=[(i, i) for i in range(1, 6)])
    comment = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return f'{self.client.username} - {self.service.title} ({self.rating} stars)'
```

**urls.py**

```
from django.urls import path
from .views import service_reviews, add_review, edit_review, delete_review,
view_service_reviews, provider_reviews
urlpatterns = [
    path('service/<int:service_id>', service_reviews, name="service_reviews"),
    path('service/<int:service_id>/add/', add_review, name="add_review"),
    path('review/<int:review_id>/edit/', edit_review, name="edit_review"),
    path('review/<int:review_id>/delete/', delete_review, name="delete_review"),
    path('service/<int:service_id>/reviews/', view_service_reviews,
name="view_service_reviews"),
    path('provider/reviews/', provider_reviews, name="provider_reviews"),
]
```

**views.py**

```

from django.shortcuts import render, get_object_or_404, redirect
from django.contrib.auth.decorators import login_required
from django.http import HttpResponseRedirect
from .models import Review
from services.models import Service
from bookings.models import Booking
def service_reviews(request, service_id):
    service = get_object_or_404(Service, id=service_id)
    reviews = Review.objects.filter(service=service).order_by('-created_at')
    return render(request, "reviews/service_reviews.html", {"service": service, "reviews": reviews})
@login_required
def add_review(request, service_id):
    service = get_object_or_404(Service, id=service_id)
    if not Booking.objects.filter(client=request.user, service=service, status="Completed").exists():
        return redirect("service_reviews", service_id=service.id)
    existing_review = Review.objects.filter(client=request.user, service=service).first()
    if existing_review:
        return redirect("edit_review", review_id=existing_review.id)
    if request.method == "POST":
        rating = request.POST.get("rating")
        comment = request.POST.get("comment")
        try:
            rating = int(rating)
            if rating < 1 or rating > 5:
                raise ValueError
        except (TypeError, ValueError):
            return render(request, "reviews/review_form.html", {"service": service, "error": "Invalid rating!"})
        if rating and comment:
            Review.objects.create(client=request.user, service=service, rating=rating, comment=comment)
            return redirect("profile")
        return render(request, "reviews/review_form.html", {"service": service})
@login_required
def edit_review(request, review_id):
    review = get_object_or_404(Review, id=review_id, client=request.user)
    if request.method == "POST":
        review.rating = request.POST.get("rating")
        review.comment = request.POST.get("comment")
        review.save()
        return redirect("profile")
    return render(request, "reviews/edit_review.html", {"review": review})
@login_required
def delete_review(request, review_id):

```

```

review = get_object_or_404(Review, id=review_id, client=request.user)
if request.method == "POST":
    service_id = review.service.id
    review.delete()
    return redirect("profile")
return render(request, "reviews/delete_review.html", {"review": review})
@login_required
def view_service_reviews(request, service_id):
    service = get_object_or_404(Service, id=service_id)
    reviews = Review.objects.filter(service=service).order_by('-created_at')
    return render(request, "reviews/view_service_reviews.html", {"service": service, "reviews": reviews})
@login_required
def provider_reviews(request):
    provider_services = Service.objects.filter(provider=request.user)
    provider_reviews = Review.objects.filter(service__in=provider_services).order_by('-created_at')
    return render(request, "reviews/provider_reviews.html", {"provider_reviews": provider_reviews})

```

**static****css/review\_form\_style.css**

```

.container{max-width:650px;background:#fff;padding:25px;border-radius:10px;box-shadow:0 4px 12px rgba(0,0,0,0.1);}
h2{color:#0078E8;font-weight:bold;text-align:center;margin-bottom:20px;}
.card{background:#f8f9fa;border-radius:8px;padding:15px;box-shadow:0 2px 5px rgba(0,0,0,0.08);}
.card h5,.card p{margin-bottom:5px;}
.service-img{width:100%;max-height:150px;object-fit:cover;border-radius:8px;}
.provider-img{width:60px;height:60px;object-fit:cover;border-radius:50%;border:2px solid #0078E8;}
.rating{display:flex;flex-direction:row-reverse;justify-content:start;font-size:2.2rem;gap:5px;}
.rating input{display:none;}
.rating label{cursor:pointer;color:#bbb;transition:color 0.3s ease,transform 0.2s ease-in-out;}
.rating input:checked~label,.rating label:hover,.rating label:hover~label{color:#FFD700;transform:scale(1.1);}
.rating input:checked~label{text-shadow:0 0 10px rgba(255,215,0,0.7);}
.rating label:hover{text-shadow:0 0 8px rgba(255,215,0,0.5);}
.form-label{font-weight:bold;color:#333;}
textarea{border-radius:5px;resize:none;transition:all 0.2s;}
textarea:focus{border-color:#0078E8;box-shadow:0 0 5px rgba(0,120,232,0.5);}
.btn-group{display:flex;gap:10px;justify-content:center;}
.btn-primary,.btn-secondary{width:160px;font-size:1rem;font-weight:bold;border-radius:6px;padding:10px 0;text-align:center;transition:all 0.3s ease-in-out;}
.btn-primary{background:linear-gradient(135deg,#0078E8 0%,#005bbf 100%);color:#fff;border:none;}

```

```

.btn-primary:hover{background:linear-gradient(135deg,#005bbf 0%,#0047a3
100%);transform:scale(1.05);box-shadow:0 4px 10px rgba(0,120,232,0.3);}
.btn-secondary {background:linear-gradient(135deg,#6c757d 0%,#5a6268
100%);color:#fff; border:none;}
.btn-secondary:hover{background:linear-gradient(135deg,#5a6268 0%,#484e53
100%);transform:scale(1.05);box-shadow:0 4px 10px rgba(108,117,125,0.3);}
@media(max-width:768px){.container {max-width:100%;padding:15px;}.rating {font-
size:1.8rem;}.btn-group {flex-direction:column;align-items:center;}.btn-primary,.btn-
secondary {width:100%;}}
css/review_list_style.css
body {background:linear-gradient(to right,#f4f6f9,#dbe2ef);}

.main-card {background:rgba(255,255,255,0.95);border-radius:12px;padding:25px;box-
shadow:0px 10px 30px rgba(0,0,0,0.1);backdrop-filter:blur(12px);}

.page-title {color:#0078E8;font-weight:bold;font-size:1.8rem;text-transform:uppercase;letter-
spacing:1px;}

.service-section {background:rgba(245,245,245,0.95);border-radius:10px;padding:20px;box-
shadow:inset 0px 2px 10px rgba(0,0,0,0.08);transition:transform 0.3s ease-in-out;}

.service-section:hover {transform:scale(1.02);}

.service-img {max-width:100%;height:auto;border-radius:10px;box-shadow:0px 4px 10px
rgba(0,0,0,0.2);}

.service-title {font-size:1.5rem;color:#0078E8;font-weight:bold;}

.price {font-size:1.3rem;color:#28a745;font-weight:bold;}

.divider {margin:25px 0; border:none; height:3px; background:linear-gradient(to
right,#0078E8,#28a745);border-radius:5px; }

.review-section {background:rgba(255,255,255,0.9);padding:20px;border-radius:10px;box-
shadow:inset 0px 2px 10px rgba(0,0,0,0.05);}

.review-card {background:rgba(245,245,245,0.9);padding:15px;border-radius:12px; border-
left:6px solid #0078E8; margin-bottom:15px;box-shadow:0px 3px 15px
rgba(0,0,0,0.08);transition:transform 0.3s ease-in-out; }

.review-card:hover {transform:scale(1.02);}

.stars {font-size:1.3rem;color:gold; }

.star {color:#ccc; }

.filled {color:gold; }

.review-header {display:flex;justify-content:space-between;align-items:center; border-bottom:1px
solid #ddd; padding-bottom:5px; margin-bottom:10px; }

.client-name {font-weight:bold;color:#0078E8; }

.review-date {font-size:0.9rem;color:#555; }

.review-comment {font-size:1rem;color:#333; }

.btn-back {background-color:#0078E8;color:white;padding:12px 25px;font-size:1.1rem; border-
radius:8px;text-decoration:none;transition:all 0.3s ease-in-out; }

.btn-back:hover {background-color:#005bb5;transform:scale(1.05); }

```

**templates****reviews/review\_form.html**

```

{%
  extends "home_base.html"
  %}
{%
  load static
  %}
{%
  block title
  %} Add Review | {{ service.title }} {%
  endblock
  %}

```

```

{% block style %}
<link rel="stylesheet" href="{% static 'css/review_form_style.css' %}">
{% endblock %}
{% block content %}
<br><br>
<div class="container mt-4">
    <h2 class="page-title text-center">📝 Add Your Review</h2>
    <div class="card p-3 mb-4 service-details">
        <div class="row">
            <div class="col-md-4 text-center">
                {% if service.image %}
                    
                {% endif %}
            </div>
            <div class="col-md-8">
                <h5 class="service-title">{{ service.title }}</h5>
                <p><strong>Provider:</strong> {{ service.provider.username }}</p>
                <p><strong>Category:</strong> {{ service.category }}</p>
                <p><strong>Price:</strong> <span class="price">₹{{ service.price }}</span></p>
                <p><strong>Description:</strong> {{ service.description }}</p>
                <p><strong>Created On:</strong> {{ service.created_at|date:"M d, Y" }}</p>
            </div>
        </div>
    </div>
    <form method="POST" class="review-form">
        {% csrf_token %}
        <div class="mb-3">
            <label class="form-label"><strong>Rating</strong></label>
            <div class="rating">
                {% for i in "54321"|make_list %}
                    <input type="radio" id="star{{ i }}" name="rating" value="{{ i }}" required>
                    <label for="star{{ i }}" title="{{ i }} stars">★</label>
                {% endfor %}
            </div>
        </div>
        <div class="mb-3">
            <label for="comment" class="form-label"><strong>Comment</strong></label>
            <textarea name="comment" id="comment" class="form-control" rows="4" required></textarea>
        </div>
        <div class="btn-group">
            <button type="submit" class="btn btn-primary">✓ Submit Review</button>
            <a href="{% url 'profile' %}" class="btn btn-secondary">✗ Cancel</a>
        </div>
    </form>

```

```

</div>
{% endblock %}
reviews/service_reviews.html
{% extends "base.html" %}
{% block content %}
<div class="container">
    <h2>Reviews for {{ service.title }}</h2>
    <a href="{% url 'add_review' service.id %}" class="btn btn-primary">Add Review</a>
    <ul class="list-group mt-3">
        {% for review in reviews %}
        <li class="list-group-item">
            <strong>{{ review.client.username }}</strong> - {{ review.rating }} ★
            <p>{{ review.comment }}</p>
            <small>{{ review.created_at }}</small>
        </li>
        {% empty %}
        <li class="list-group-item">No reviews yet.</li>
        {% endfor %}
    </ul>
</div>
{% endblock %}
reviews/view_service_review.html
{% extends "home_base.html" %}
{% load static %}
{% block title %} Reviews | {{ service.title }}{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/review_list_style.css' %}">
{% endblock %}
{% block content %}
<div class="container mt-5">
    <br>
    <div class="card main-card shadow-lg p-4">
        <h2 class="page-title">★ Reviews for "{{ service.title }}"</h2>
        <div class="service-section">
            <div class="row align-items-center">
                <div class="col-md-4 text-center">
                    {% if service.image %}
                        
                    {% endif %}
                </div>
                <div class="col-md-8">
                    <h4 class="service-title">{{ service.title }}</h4>
                    <p><i class="fas fa-user-tie"></i> <strong>Provider:</strong> {{ service.provider.username }}</p>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<p><i class="fas fa-tag"></i> <strong>Category:</strong> {{ service.category
}}</p>
<p><i class="fas fa-rupee-sign"></i> <strong>Price:</strong> <span
class="price">₹{{ service.price }}</span></p>
<p><i class="fas fa-info-circle"></i> <strong>Description:</strong> {{
service.description }}</p>
<p><i class="fas fa-calendar-alt"></i> <strong>Created On:</strong> {{
service.created_at|date:"M d, Y" }}</p>
</div>
</div>
</div>
<hr class="divider">
<div class="review-section">
<h4 class="review-heading">⭐ User Reviews</h4>
<br>
{% if reviews %}
  {% for review in reviews %}
    <div class="review-card">
      <div class="review-header">
        <p class="client-name"><i class="fas fa-user"></i> <strong>Client:</strong>
{{ review.client.username }}</p>
        <p class="review-date text-muted"><i class="fas fa-clock"></i> {{
review.created_at|date:"M d, Y" }} at {{ review.created_at|time:"h:i A" }}</p>
      </div>
      <p><strong>Rating:</strong>
        <span class="stars">
          {% for i in "12345"|make_list %}
            {% if forloop.counter <= review.rating %}
              <span class="star filled">★</span>
            {% else %}
              <span class="star">☆</span>
            {% endif %}
          {% endfor %}
        </span> ({{ review.rating }}/5)
      </p>
      <p class="review-comment"><i class="fas fa-comment-dots"></i>
<strong>Comment:</strong> {{ review.comment }}</p>
    </div>
  {% endfor %}
  {% else %}
    <p class="text-muted">🚫 No reviews yet.</p>
  {% endif %}
</div>
<div class="mt-4">
  <a href="{{ url 'profile' }}><i class="fas fa-arrow-left"></i>
Back to Profile</a>

```

```

        </div>
    </div>
    <br><br><br>
</div>
{% endblock %}
reviews/edit_review.py
{% extends "home_base.html" %}
{% load static %}
{% block title %} Edit Review | {{ review.service.title }}{% endblock %}
{% block style %}
<link rel="stylesheet" href="{% static 'css/review_form_style.css' %}">
{% endblock %}
{% block content %}
<br><br>
<div class="container mt-4">
    <h2 class="page-title text-center">✍ Edit Your Review</h2>
    <div class="card p-3 mb-4 service-details">
        <div class="row">
            <div class="col-md-4 text-center">
                {% if review.service.image %}
                    
                {% endif %}
            </div>
            <div class="col-md-8">
                <h5 class="service-title">{{ review.service.title }}</h5>
                <p><strong>Provider:</strong> {{ review.service.provider.username }}</p>
                <p><strong>Category:</strong> {{ review.service.category }}</p>
                <p><strong>Price:</strong> <span class="price">₹{{ review.service.price }}</span></p>
                <p><strong>Description:</strong> {{ review.service.description }}</p>
                <p><strong>Created On:</strong> {{ review.service.created_at|date:"M d, Y" }}</p>
            </div>
        </div>
    </div>
    <form method="POST" class="review-form">
        {% csrf_token %}
        <div class="mb-3">
            <label class="form-label"><strong>Rating</strong></label>
            <div class="rating">
                {% for i in "54321"|make_list %}
                    <input type="radio" id="star{{ i }}" name="rating" value="{{ i }}" {% if review.rating == i|add:0 %}checked{% endif %} required>
                    <label for="star{{ i }}" title="{{ i }} stars">★</label>
                {% endfor %}
            </div>
        </div>
    </form>
</div>

```

```

</div>
<div class="mb-3">
    <label for="comment" class="form-label"><strong>Comment</strong></label>
    <textarea name="comment" id="comment" class="form-control" rows="4" required>{{ review.comment }}</textarea>
</div>
<div class="btn-group">
    <button type="submit" class="btn btn-primary"> Update Review</button>
    <a href="{% url 'profile' %}" class="btn btn-secondary"> Cancel</a>
</div>
</form>
</div>
{% endblock %}
reviews/delete_review.html
{% extends "base.html" %}
{% block content %}
<div class="container mt-4">
    <h2>Delete Review</h2>
    <div class="card p-3 mb-4">
        <h5><strong>Service:</strong> {{ review.service.title }}</h5>
        <p><strong>Provider:</strong> {{ review.service.provider.username }}</p>
        <p><strong>Client:</strong> {{ review.client.username }}</p>
    </div>
    <div class="alert alert-warning">
        <p>Are you sure you want to delete this review?</p>
        <p><strong>Rating:</strong> {{ review.rating }} 

```

## 11. APPLICATIONS

1. **Online Tutoring Platform:** EduMart allows tutors and subject-matter experts to offer virtual lessons to students across academic levels and subjects. Students can browse profiles, view qualifications, book sessions, and leave feedback.
2. **Skill-Based Training Hub:** Individuals seeking to learn technical or creative skills like coding, graphic design, photography, or music can use EduMart to find skilled trainers and enroll in one-on-one or group sessions.
3. **Freelance Educational Services:** Freelancers offering educational services such as resume writing, academic content creation, assignment help, or career counseling can use EduMart to showcase their offerings and manage client interactions.
4. **Portfolio Showcasing for Educators:** Educators and trainers can build and display a verified portfolio on EduMart, including reviews, ratings, and credentials, helping them gain credibility and attract more clients.
5. **Centralized Booking & Payment System:** The platform provides a one-stop solution for discovering, booking, and paying for services—eliminating the need for third-party tools. It's especially useful for managing session-based services.
6. **Localized Educational Services:** Users can discover and connect with local service providers in their area, supporting in-person sessions for subjects like home tutoring, language coaching, or exam prep.
7. **Student Support Services:** EduMart can be used by academic support professionals—such as test prep coaches, special education tutors, or language translators—to connect with students who need specific educational assistance.
8. **Educational Startups & Institutions:** Small coaching centers or educational startups can use EduMart as a marketing and booking tool to reach a broader audience and manage inquiries, enrollments, and payments.
9. **Remote Learning for Underserved Areas:** EduMart can bridge the educational gap in rural or underserved regions by allowing access to skilled educators remotely, enhancing inclusivity and learning opportunities.
10. **Career & Personal Development:** Users can find services that focus on soft skills, personal growth, and career development, such as interview training, public speaking coaching, or time management sessions.

## 12. CONCLUSION

**EduMart** successfully establishes itself as a comprehensive and user-friendly online marketplace for educational and skill-based services. By bridging the gap between clients in need of educational support and professionals offering specialized services, **EduMart** creates a reliable platform for seamless interaction, booking, and learning.

The project ensures that both clients and service providers have a secure, transparent, and efficient environment to collaborate. Features like real-time bookings, secure payments, messaging, ratings & reviews, and categorized service listings collectively enhance the overall user experience and build trust on the platform.

From an implementation perspective, the use of Django as the backend framework, SQLite for lightweight yet efficient data handling, and modern frontend technologies has allowed the platform to be scalable, maintainable, and performance-driven. The integration of a custom admin panel and session-based middleware further strengthens the project's focus on user management and security.

In essence, **EduMart** not only addresses the limitations of traditional educational service discovery methods but also sets a strong foundation for future expansion and real-world deployment. It stands as a promising solution for democratizing access to educational resources and empowering professionals to reach their target audience effectively.

## 13. FUTURE SCOPE

EduMart has been developed as a foundational platform to connect clients and educational service providers. However, there remains significant potential to expand and enhance its capabilities. The following future improvements can be explored to further enrich the platform.

1. **Mobile Application Development:** A dedicated mobile app for Android and iOS can be developed to improve accessibility and user engagement, offering push notifications, instant chat, and a more intuitive interface on the go.
2. **Video Calling Integration:** Integrating a secure video conferencing feature will allow service providers to conduct live sessions or consultations directly through the platform, improving real-time interactions.
3. **AI-Based Service Recommendations:** Implementing machine learning algorithms to recommend personalized services based on user preferences, past bookings, and search behavior can enhance user experience and platform retention.
4. **Multi-Language Support:** Adding support for regional and international languages can expand the platform's reach and make it more inclusive for users from different linguistic backgrounds.
5. **Subscription-Based Plans for Providers:** Introducing tiered subscription models with added benefits (e.g., priority listing, analytics, more service slots) can help monetize the platform and offer better visibility to premium service providers.
6. **Advanced Analytics Dashboard:** A detailed analytics module for both clients and providers can be introduced to track performance, feedback, booking trends, and income reports for better decision-making.
7. **Gamification and Rewards:** Adding a reward or badge system for top-rated service providers and frequent clients can increase platform activity, engagement, and loyalty.
8. **Integration with External Tools:** Integration with tools like Google Calendar, Zoom, or online payment wallets (UPI, Paytm, Google Pay) can improve user convenience and automation.
9. **Support for Digital Content Sales:** Allowing providers to sell digital materials such as eBooks, video tutorials, or recorded classes directly from their profiles can open up new revenue streams.
10. **Automated Dispute Resolution System:** Implementing an AI-assisted or rule-based system to handle booking disputes and refund claims can improve transparency and reduce manual admin effort.

## 14. MILESTONES

<b>Sr. No</b>	<b>Project activity</b>	<b>Estimated Start Date</b>	<b>Estimated End Date</b>
1.	Project Allotment	08/01/2025	11/01/2025
2.	Requirement Gathering	11/01/2025	15/01/2025
3.	Synopsis Creation	15/01/2025	23/01/2025
4.	Synopsis Submission	24/01/2025	24/01/2025
5.	Designing	25/01/2025	05/02/2025
6.	Development	06/02/2025	28/02/2025
7.	Testing	01/03/2025	07/03/2025
8.	Finalizing the Project	08/03/2025	12/03/2025

## 15. MEETINGS WITH THE SUPERVISOR

Date of Meet	Comments by the Supervisor	Signature of the Supervisor
10/01/2025	Project topic discussions	
06/08/2024	Synopsis Correction and Model Creation	
22/01/2025	Corrections in Synopsis	
23/01/2025	Checked the revised Synopsis and approved for presentation	
18/02/2025	First demonstration of working project (Login and Registration part)	
03/03/2025	Demonstration of Service and Booking Module	
18/03/2025	Demonstration of Payment Module	
04/04/2025	Final demonstration of complete Project	
07/04/2025	Checked the Project Report and approved for presentation	
	Checked the Project Report and approved for final presentation	

## 16. REFERENCES

- [1] GeeksforGeeks, "Levels in Data Flow Diagrams (DFD)," *GeeksforGeeks*, [Online]. Available: <https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/>
- [2] diagrams.net, "Diagrams.net (Formerly Draw.io)," *diagrams.net*, [Online]. Available: <https://app.diagrams.net/>
- [3] Bootstrap, "Bootstrap – Front-End Framework," *getbootstrap.com*, [Online]. Available: <https://getbootstrap.com/>
- [4] Django Software Foundation, "Django – Python Web Framework," *Django Official Website*, [Online]. Available: <https://www.djangoproject.com/start/>
- [5] Mozilla Developer Network (MDN), "MDN Web Docs – Developer Documentation," *developer.mozilla.org*, [Online]. Available: <https://developer.mozilla.org/en-US/>
- [6] W3Schools, "Web Development Tutorials," *W3Schools*, [Online]. Available: <https://w3schools.com/>
- [7] Unsplash, "Free Image Repository," *Unsplash*, [Online]. Available: <https://unsplash.com/>
- [8] Color Hunt, "Color Palettes for Design," *Color Hunt*, [Online]. Available: <https://colorhunt.co/>
- [9] Razorpay, "Payment Gateway for India - Best Payment Gateway for Online Payments," *Razorpay*, [Online]. Available: <https://razorpay.com>. [Accessed: 07-Apr-2025].
- [10] Freelancer.com, "Freelance Jobs and Freelancers," *Freelancer*, [Online]. Available: <https://www.freelancer.com>. [Accessed: 07-Apr-2025].

## 17. PLAGIARISM REPORT

 April 04, 2025

### Plagiarism Scan Report

2% Plagiarized | 98% Unique

Characters: 7263 | Words: 973  
Sentences: 53 | Speak Time: 8 Min

2% Exact Matched | 0% Partial Matched

Excluded URL: None

#### Content Checked for Plagiarism

INTRODUCTION EduBazar is a dynamic web-based platform designed to connect individuals seeking educational services with skilled professionals who offer them. The platform enables clients to discover, book, and procure a diverse range of services such as subject tutorials, graphic designing, application development, and more. At the same time, it empowers service providers to showcase their expertise, manage their service listings, communicate with clients, and handle bookings efficiently. The project focuses on creating a user-friendly, secure, and scalable web application that facilitates smooth interactions between service providers and clients. EduBazar incorporates modern features including real-time chat, booking and payment integration, and a review and rating system, ensuring a seamless service experience. The platform uses Django for the backend, HTML/CSS/JS for the frontend, and SQLite as the database. By offering

web browsers like Chrome, Firefox, Brave, and Safari, and is compatible with Windows, macOS, and Linux. • Maintainability: Clean, modular Django code and comprehensive inline documentation enable developers to easily maintain and extend the platform. • Backup and Data Recovery: Regular database backups are recommended for production deployment to prevent data loss in case of system failure. • Accessibility: The platform follows basic accessibility guidelines, with support for keyboard navigation and screen readers where applicable.

Technical Requirements 1. Frontend Technologies • HTML: Creating Structure and markup for webpages. • CSS: Styling and layout using Custom CSS. • JavaScript: Client-side scripting for interactive features and game logic. • Bootstrap: Framework for more prebuilt features for defining interactive UI. 2. Backend Technologies • Django: Framework for building the backend, handling routing, and interacting with the database. • Database: SQLite for storing user data, game data, and performance metrics. • ORM: Django's ORM (Object Relational Mapping) for database interactions.

#### Sources

##### 100% Plagiarized

User Profiles: Each user has a dedicated profile and resume. Job Posting & Discovery: Users can post job roles and find relevant job opportunities. Project Sharing: Users can upload and view projects for collaboration and inspiration. Real-time Updates: Seamless data synchronization using Firebase.

[github.com/Kalamatha-Eshwari/Let-s-Find](https://github.com/Kalamatha-Eshwari/Let-s-Find)



[Home](#) [Blog](#) [Testimonials](#) [About Us](#) [Privacy Policy](#)

Copyright © 2025 **Plagiarism Detector**. All right reserved

providers. 2. Enhanced User Experience: Create a user-friendly interface for seamless navigation and bookings. 3. Trust and Transparency: Implement reviews and ratings to ensure accountability and quality services. 4. Empower Providers: Enable professionals to showcase expertise and manage bookings efficiently. 5. Streamlined Processes: Simplify service discovery and bookings with advanced tools and real-time updates. 6. Security: Protect user data and payments with secure authentication and encryption. 7. Customization: Offer personalized profiles and tailored listings for better user relevance. 8. Scalability: Design the platform to handle growth without compromising performance. 9. Real-Time Communication: Integrate messaging for seamless client-provider interactions. 10. Continuous Improvement: Monitor feedback and update features to enhance functionality. REQUIREMENT ANALYSIS AND SPECIFICATIONS Functional Requirements • User Registration and Authentication: EduBazar allows users to register as either clients or service providers. Secure login and registration are implemented using email and password authentication. Each user can manage their profile details. • Service Listings: Service providers can create and manage listings by adding titles, descriptions, categories (like tutorials, design, or development), pricing, and availability. Listings are stored and updated dynamically in the database. • Advanced Search and Filters: Clients can search for services using various filters such as category, pricing, availability, and user ratings to find the most