

# **Final Report**

## **EarnSync- Freelance Portal**

### **Deliverable 03**

#### **Team Members:**

Muhammad Bilal Khawar (22i-0786)  
Saffi M.Hashir (22i-1293)  
Hamza Ahmed Siddique (22i-1339)

#### **Submission Date:**

April 27, 2025

---

# Table of Contents

<b>1. Introduction.....</b>	<b>4</b>
1.1 About the Project.....	4
1.2 Core Goals.....	4
1.3 Key User Roles.....	4
1.4 System Highlights.....	5
<b>2. Functional Requirements.....</b>	<b>5</b>
2.1 User Management:.....	5
2.2 Freelancer Capabilities:.....	5
2.3 Employer Capabilities:.....	5
2.4 Payment System:.....	6
2.5 Administration:.....	6
2.6 Rating and Reviews:.....	6
2.7 Communication:.....	7
<b>3. Non-Functional Requirements.....</b>	<b>7</b>
3.1 Product Requirements:.....	7
3.2 Organizational Requirements:.....	8
3.3 External Requirements:.....	8
<b>4. User Stories with Pre and Post Conditions.....</b>	<b>9</b>
<b>5. Product Backlog.....</b>	<b>12</b>
5.1 High Priority User Stories.....	12
5.2 Medium Priority User Stories.....	12
5.3 Low Priority User Stories.....	12
<b>6. Sprint Backlog.....</b>	<b>13</b>
6.1 Sprint 1 Backlog.....	13
6.2 Sprint 2 Backlog.....	14
<b>7. Project Plan.....</b>	<b>15</b>
7.1 Project Overview.....	15
7.2 Project Objectives.....	15
7.3 Work Breakdown Structure (WBS).....	16
7.4 Project Gantt Chart.....	18
7.5 Roles and Responsibilities.....	18
7.6 Technology Stack.....	18
<b>8. System Architecture.....</b>	<b>19</b>
8.1 Use Case Diagram.....	19
8.2 Class Diagram.....	20
8.3 Package Diagram.....	21
8.4 Architecture Diagram.....	23
8.5 Deployment Diagram.....	24

8.6 Component Diagram.....	25
<b>9. Sprint 3 Items.....</b>	<b>25</b>
<b>10. Implementation Screenshots.....</b>	<b>29</b>
<b>11. Burndown chart.....</b>	<b>41</b>
<b>12. Trello Board Screenshots.....</b>	<b>41</b>
<b>13. TestCases - Black Box.....</b>	<b>43</b>
13.1 Equivalence Class Partitioning.....	43
13.2 Boundary Value Analysis.....	44
<b>14. TestCases - White Box.....</b>	<b>45</b>
14.1 Tools Used.....	45
14.2 Features Tested.....	45
14.3 Code Coverage Summary.....	46
14.4 Results.....	46
14.5 What Is Covered vs Not Covered.....	47
<b>15. Work Division between Group Members.....</b>	<b>48</b>
<b>16. Lessons Learnt by Group.....</b>	<b>48</b>

# 1. Introduction

## 1.1 About the Project

**EarnSync** is a comprehensive freelance job marketplace designed to streamline collaboration between **freelancers** and **employers**, while providing **administrative control** to ensure quality, fairness, and security.

The platform supports the **entire lifecycle** of a freelance engagement — from job posting and proposal submission to milestone-based tracking, payment handling, and dispute resolution. Built with **modular microservices** and cleanly separated frontends, the system provides scalability, maintainability, and flexibility.

## 1.2 Core Goals

- **Enable freelancers** to find and apply for jobs, manage milestones, and receive payments.
- **Empower employers** to post jobs, review applications, and control project progress.
- **Equip admins/super admins** with dashboards, dispute resolution tools, and analytics to ensure a smooth user experience.
- **Implement secure, efficient, and scalable microservices** to handle authentication, payments, messaging, and job management.

## 1.3 Key User Roles

Role	Responsibilities
<b>Freelancer</b>	Create profile, search jobs, apply, manage milestones, withdraw earnings
<b>Employer</b>	Post jobs, review proposals, assign projects, track milestones, process payments
<b>Admin</b>	Handle disputes, verify users, perform soft deletes, access dashboards
<b>Super Admin</b>	Manage admins, view system-wide stats, generate reports

## **1.4 System Highlights**

- **Milestone-based Project Flow** – Employers create and fund milestones; freelancers submit deliverables for each.
- **Role-Based Access Control** – Ensures distinct permissions and views for each user role.
- **Integrated Wallet System** – Freelancers can withdraw payments; employers fund milestones securely.
- **Real-Time Chat & Dispute Resolution** – Integrated communication and admin-intervened dispute handling.
- **Admin & Super Admin Dashboards** – Track KPIs, generate reports, and access dispute histories.

## **2. Functional Requirements**

### **2.1 User Management:**

- The system shall implement role-based access control (freelancer, employer, admin).
- The system shall allow users to reset passwords via email verification.
- The system shall support profile deactivation and deletion.

### **2.2 Freelancer Capabilities:**

- The system shall allow freelancers to create detailed profiles with:
  - Personal information (name, contact details)
  - Professional skills and expertise
  - Educational background
  - Work experience
  - Hourly rate or compensation expectations
  - Availability status
- The system shall enable freelancers to browse, search, and filter job listings.
- The system shall allow freelancers to apply to jobs with cover letters.

## **2.3 Employer Capabilities:**

- The system shall allow employers to post job listings with:
  - Job title and description
  - Required skills and experience
  - Project type (hourly, fixed price)
  - Budget range or hourly rate
  - Timeline/deadline
- The system shall allow employers to edit or remove job postings.
- The system shall allow employers to search and filter freelancer profiles.
- The system shall allow employers to shortlist freelancers for consideration.
- The system shall enable employers to review applications and select freelancers.
- The system shall allow employers to create milestones for projects.
- The system shall enable employers to release payments upon:
  - Milestone completion
  - Project completion
  - Approved hourly work logs

## **2.4 Payment System:**

- The system shall integrate with secure payment gateways to facilitate transactions.
- The system shall support escrow functionality to hold payments until work is approved.
- The system shall calculate and deduct platform fees from transactions.
- The system shall generate invoices for completed payments.
- The system shall maintain transaction history for all users.

## **2.5 Administration:**

- The system shall allow admins to monitor platform activity.
- The system shall provide admins with tools to:
  - Verify user identities when required
  - Review reported content or users
  - Resolve disputes between freelancers and employers
  - Moderate job postings and profiles
- The system shall enable admins to suspend or ban users for violations.
- The system shall generate reports on platform usage and transactions.

## **2.6 Rating and Reviews:**

- The system shall allow employers to rate and review freelancers after project completion.
- The system shall allow freelancers to rate and review employers after project completion.
- The system shall display average ratings on user profiles.

## **2.7 Communication:**

- The system shall provide a messaging system for freelancers and employers to communicate.
- The system shall support file attachments in messages.
- The system shall notify users of new messages via email or in-app notifications.
- The system shall maintain message history for the duration of the project and a defined period after completion.

# **3. Non-Functional Requirements**

## **3.1 Product Requirements:**

### **A. Performance:**

- The system shall support at least 10,000 concurrent users.
- The system shall provide response times of less than 2 seconds for common operations.
- The system shall be available 99.5% of the time (excluding scheduled maintenance).

### **B. Security:**

- The system shall encrypt all sensitive data at rest and in transit.
- The system shall implement multi-factor authentication for account access.
- The system shall implement protection against common web vulnerabilities (XSS, CSRF, SQL injection).
- The system shall maintain audit logs for security-relevant events.

### **C. Reliability:**

- The system shall perform daily backups of all data.
- The system shall implement disaster recovery procedures with a recovery time objective of 4 hours.

- The system shall gracefully degrade functionality during peak loads rather than crash.

**D. Usability:**

- The system shall be accessible on desktop and mobile devices.
- The system shall comply with WCAG 2.1 level AA accessibility guidelines.
- The system shall support internationalization and localization for at least 3 languages.
- The system shall provide intuitive navigation with no more than 3 clicks to reach any major function.

**E. Maintainability:**

- The system shall follow a modular design to facilitate future enhancements.
- The system shall include comprehensive documentation for maintenance and support.
- The system shall support updates and patches without extended downtime.

**F. Scalability:**

- The system shall be designed to scale horizontally to accommodate growth.
- The system shall handle a 300% increase in users without significant performance degradation.

**3.2 Organizational Requirements:**

- The system shall follow a modular design to facilitate future enhancements.
- The system shall include comprehensive documentation for maintenance and support.
- The system shall support updates and patches without extended downtime.

**3.3 External Requirements:**

- The system shall ensure adherence to applicable data protection and privacy standards.
- The system shall be designed with inclusive and accessible user experiences in mind.
- The system shall support integration with external communication channels for user notifications.

## 4. User Stories with Pre and Post Conditions

Each of the following user stories shall be divided into two tasks, frontend and backend.

1. **As a Freelancer, I want to search and apply for jobs**, so that I can find work opportunities that match my skills and earn income.
  - **Pre-conditions:** Freelancer is registered and logged in; job postings are available on the platform.
  - **Post-conditions:** Freelancer views job listings and successfully submits applications.
2. **As a Freelancer, I want to communicate with employers and admins**, so that I can collaborate effectively and resolve any project-related queries or issues.
  - **Pre-conditions:** Freelancer is logged in; a project or job-related conversation thread exists or can be initiated.
  - **Post-conditions:** Messages are sent and stored in the system; employer/admin can view and respond.
3. **As an Employer, I want to communicate with freelancers and admins**, so that I can manage project progress and clarify job requirements or concerns.
  - **Pre-conditions:** Employer is logged in; freelancer or admin is available for communication.
  - **Post-conditions:** Communication is recorded; queries are addressed.
4. **As an Admin, I want to communicate with freelancers and employers**, so that I can support users and resolve platform-related issues.
  - **Pre-conditions:** Admin is logged in; a user has initiated a support request or conversation.
  - **Post-conditions:** Support is provided; issues are resolved or escalated.
5. **As a Freelancer, I want to track my earnings and generate financial reports**, so that I can monitor my income and manage my finances.
  - **Pre-conditions:** Freelancer is logged in; freelancer has completed paid tasks or projects.
  - **Post-conditions:** Reports are generated and displayed or downloaded by the user.

6. **As a Freelancer, I want to raise disputes**, so that I can ensure fair treatment and project integrity when conflicts arise.
  - **Pre-conditions:** Freelancer is logged in; a project is in dispute or a conflict has occurred.
  - **Post-conditions:** Dispute ticket is created and under review by Admin.
7. **As an Employer, I want to raise disputes**, so that I can address conflicts with freelancers and ensure project success.
  - **Pre-conditions:** Employer is logged in; conflict with a freelancer has occurred.
  - **Post-conditions:** Dispute is logged and forwarded to Admin for resolution.
8. **As an Admin, I want to intervene in disputes**, so that I can mediate and ensure a fair resolution for both parties.
  - **Pre-conditions:** Admin is logged in; a dispute ticket exists in the system.
  - **Post-conditions:** Resolution is provided and case status is updated.
9. **As a Freelancer, I want to rate and review employers**, so that other freelancers can make informed decisions about job offers.
  - **Pre-conditions:** Freelancer has completed a project with an employer.
  - **Post-conditions:** Rating and review are submitted and published on employer's profile.
10. **As an Employer, I want to rate and review freelancers**, so that I can share feedback and help others hire reliable professionals.
  - **Pre-conditions:** Employer has completed a project with a freelancer.
  - **Post-conditions:** Rating and review are saved and visible on freelancer's profile.
11. **As a Freelancer, I want to register and manage my account**, so that I can maintain my personal information and access job opportunities.
  - **Pre-conditions:** Freelancer accesses registration page or profile settings.
  - **Post-conditions:** Account is created/updated and profile is saved.
12. **As an Employer, I want to register and manage my account**, so that I can post jobs and manage my hiring activities.
  - **Pre-conditions:** Employer accesses registration or account settings page.
  - **Post-conditions:** Employer account is created/updated successfully.

13. **As a Super Admin, I want to register and manage Admin accounts**, so that I can maintain platform integrity and user support.
  - **Pre-conditions:** Admin is logged in.
  - **Post-conditions:** New Admin accounts are created, updated, suspended, or deleted as needed.
14. **As an Employer, I want to post a job**, so that I can hire freelancers for specific tasks or projects.
  - **Pre-conditions:** Employer is logged in; employer account is verified.
  - **Post-conditions:** Job post is published and visible to freelancers.
15. **As an Employer, I want to review applications and select freelancers**, so that I can choose the best candidate for my job requirement.
  - **Pre-conditions:** Employer is logged in; applications for the job post are available.
  - **Post-conditions:** Freelancer is selected and assigned to the project.
16. **As an Employer, I want to create project milestones and release payments**, so that I can manage progress and ensure timely payments.
  - **Pre-conditions:** Project is in progress; employer is logged in and has linked payment details.
  - **Post-conditions:** Milestones are created; payments are released accordingly.
17. **As a Freelancer, I want to receive payments based on completed milestones**, so that I am compensated fairly and on time for my work.
  - **Pre-conditions:** Freelancer has completed the milestone task.
  - **Post-conditions:** Payment is received and reflected in the earnings dashboard.
18. **As an Admin, I want to view platform analytics and generate reports**, so that I can monitor system performance and user activity to make data-driven decisions.
  - **Pre-conditions:** Admin is logged in; data is available in the system.
  - **Post-conditions:** Reports and analytics dashboards are generated and displayed/exported.

## 5. Product Backlog

### 5.1 High Priority User Stories

- **US01 - As a Freelancer, I want to search and apply for jobs,** so that I can find work opportunities that match my skills and earn income.
- **US02 - As a Freelancer, I want to communicate with employers and admins,** so that I can collaborate effectively and resolve any project-related queries or issues.
- **US03 - As an Employer, I want to communicate with freelancers and admins,** so that I can manage project progress and clarify job requirements or concerns.
- **US11 - As a Freelancer, I want to register and manage my account,** so that I can maintain my personal information and access job opportunities.
- **US12 - As an Employer, I want to register and manage my account,** so that I can post jobs and manage my hiring activities.
- **US13 - As a Super Admin, I want to register and manage Admin accounts,** so that I can maintain platform integrity and user support.
- **US14 - As an Employer, I want to post a job,** so that I can hire freelancers for specific tasks or projects.
- **US15 - As an Employer, I want to review applications and select freelancers,** so that I can choose the best candidate for my job requirement.
- **US16 - As an Employer, I want to create project milestones and release payments,** so that I can manage progress and ensure timely payments.
- **US17 - As a Freelancer, I want to receive payments based on completed milestones,** so that I am compensated fairly and on time for my work.

### 5.2 Medium Priority User Stories

- **US04 - As an Admin, I want to communicate with freelancers and employers,** so that I can support users and resolve platform-related issues.
- **US05 - As a Freelancer, I want to track my earnings and generate financial reports,** so that I can monitor my income and manage my finances.
- **US06 - As a Freelancer, I want to raise disputes,** so that I can ensure fair treatment and project integrity when conflicts arise.
- **US07 - As an Employer, I want to raise disputes,** so that I can address conflicts with freelancers and ensure project success.
- **US08 - As an Admin, I want to intervene in disputes,** so that I can mediate and ensure a fair resolution for both parties.

### 5.3 Low Priority User Stories

- **US09 - As a Freelancer, I want to rate and review employers,** so that other freelancers can make informed decisions about job offers.
- **US10 - As an Employer, I want to rate and review freelancers,** so that I can share feedback and help others hire reliable professionals.

- **US18 - As an Admin, I want to view platform analytics and generate reports, so that I can monitor system performance and user activity to make data-driven decisions.**

## 6. Sprint Backlog

### 6.1 Sprint 1 Backlog

#### a. User System – US11,12,13 (Frontend):

Description	Freelancer/Employer Registration, Login UI
Priority	High
Assigned to	Bilal

#### b. User System – US11,12,13 (Backend):

Description	Backend APIs for auth and user management
Priority	High
Assigned to	Hashir

c. **Home Page (Frontend):**

<i>Description</i>	Homepage with system intro and basic navigation
<i>Priority</i>	Medium
<i>Assigned to</i>	Hamza

## **6.2 Sprint 2 Backlog**

a. **Admin Dashboard (Frontend):**

<i>Description</i>	Dashboard UI for admin features
<i>Priority</i>	Medium
<i>Assigned to</i>	Hamza

b. **Projects Page – US01,09,10,14,15,16 (Frontend):**

<i>Description</i>	Post jobs, apply jobs, view listings, review applications
<i>Priority</i>	High
<i>Assigned to</i>	Hamza

c. **Messages Page – US02,03,04 (Frontend):**

<i>Description</i>	Communication system between users
<i>Priority</i>	Medium
<i>Assigned to</i>	Bilal

d. **Profile Page – US11,12,13 (Frontend):**

<i>Description</i>	View/Edit Profile info (Freelancer/Employer/Admin)
<i>Priority</i>	Medium
<i>Assigned to</i>	Bilal

## 7. Project Plan

### 7.1 Project Overview

EarnSync is a full-featured freelance platform that streamlines the connection between employers and freelancers. It includes job posting, project tracking with milestones, dispute resolution, and a role-based system for freelancers, employers, and administrators.

### 7.2 Project Objectives

- Enable freelancers and employers to register, interact, and manage jobs securely.
- Allow milestone-based payments and project tracking.
- Provide a functional admin interface for oversight and moderation.
- Ensure scalability and ease of use with a clean UI/UX design.

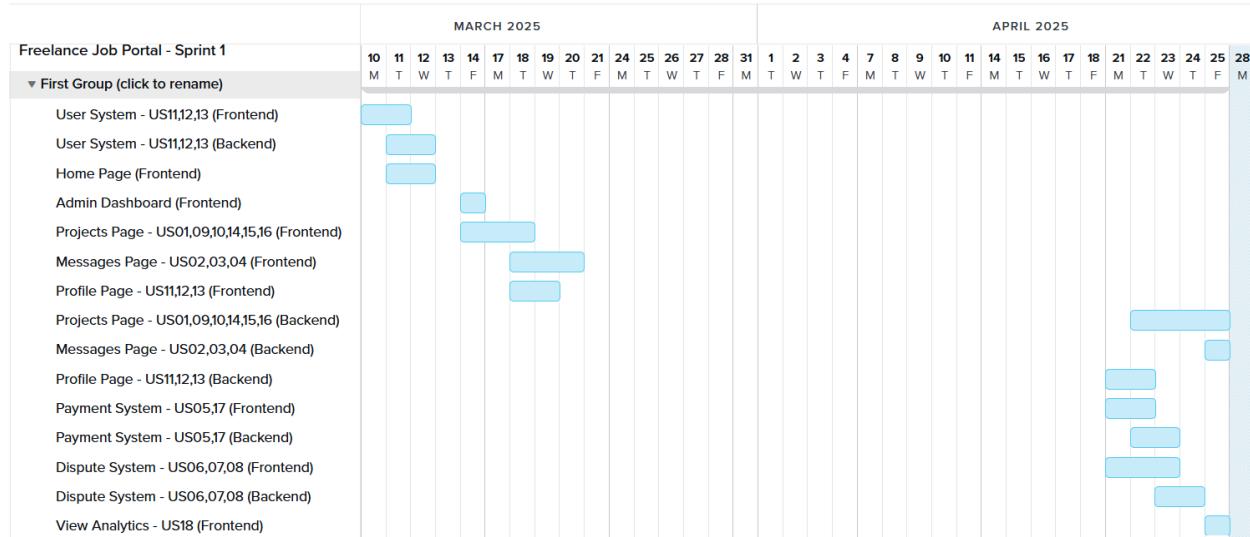
## **7.3 Work Breakdown Structure (WBS)**

### **Level 1: EarnSync Platform**

- **1.1 Sprint 1: User System and Homepage**
  - **1.1.1 User System (Frontend)**
    - 1.1.1.1 Freelancer Registration UI (US11)
    - 1.1.1.2 Employer Registration UI (US12)
    - 1.1.1.3 Login UI (US13)
  - **1.1.2 User System (Backend)**
    - 1.1.2.1 Authentication APIs (US11, US12, US13)
    - 1.1.2.2 User Management APIs (Profile Creation, Login Sessions)
  - **1.1.3 Home Page (Frontend)**
    - 1.1.3.1 System Introduction Section
    - 1.1.3.2 Basic Navigation (Login/Register Links)
- **1.2 Sprint 2: Admin, Projects, Messages, Profiles**
  - **1.2.1 Admin Dashboard (Frontend)**
    - 1.2.1.1 Admin Control Panel UI
    - 1.2.1.2 User Management Overview
  - **1.2.2 Projects Page (Frontend)**
    - 1.2.2.1 Post Jobs Form (US01)
    - 1.2.2.2 Apply for Jobs Interface (US09)
    - 1.2.2.3 View Job Listings (US10)
    - 1.2.2.4 Review Applications (US14, US15, US16)
  - **1.2.3 Messages Page**
    - 1.2.3.1 Send Messages - Backend (US02)
    - 1.2.3.2 View Messages - Backend (US03)
    - 1.2.3.3 Chat System UI (US04)
  - **1.2.4 Profile Page (Frontend)**
    - 1.2.4.1 View Profile UI (US11, US12, US13)
    - 1.2.4.2 Edit Profile Form

- **1.3 Sprint 3: Backend for Projects, Messages, Profile, Payments, Disputes, Analytics**
  - **1.3.1 Projects Page (Backend)**
    - 1.3.1.1 Job Posting APIs (US01)
    - 1.3.1.2 Job Application APIs (US09)
    - 1.3.1.3 Job Listings and Status Management APIs (US10, US14, US15, US16)
  - **1.3.2 Messages Page (Backend)**
    - 1.3.2.1 Send/Receive Message APIs (US02, US03)
    - 1.3.2.2 Chat Management APIs (US04)
  - **1.3.3 Profile Page (Backend)**
    - 1.3.3.1 Profile View/Edit APIs (US11, US12, US13)
  - **1.3.4 Payment System**
    - 1.3.4.1 Payment Initiation UI (US05)
    - 1.3.4.2 Payment Confirmation Page (US17)
    - 1.3.4.3 Transaction History UI
    - 1.3.4.4 Payment Processing Logic (US05)
    - 1.3.4.5 Payment Gateway Integration and Validations (US17)
  - **1.3.5 Dispute System**
    - 1.3.5.1 Raise Dispute UI (US06)
    - 1.3.5.2 Dispute Communication Interface (US07)
    - 1.3.5.3 Dispute History View (US08)
    - 1.3.5.4 Dispute Management APIs (US06, US07, US08)
  - **1.3.6 View Analytics**
    - 1.3.6.1 Platform Activity Dashboard UI (US18)
    - 1.3.6.2 Data Aggregation & Analytics APIs (US18)

## 7.4 Project Gantt Chart



## 7.5 Roles and Responsibilities

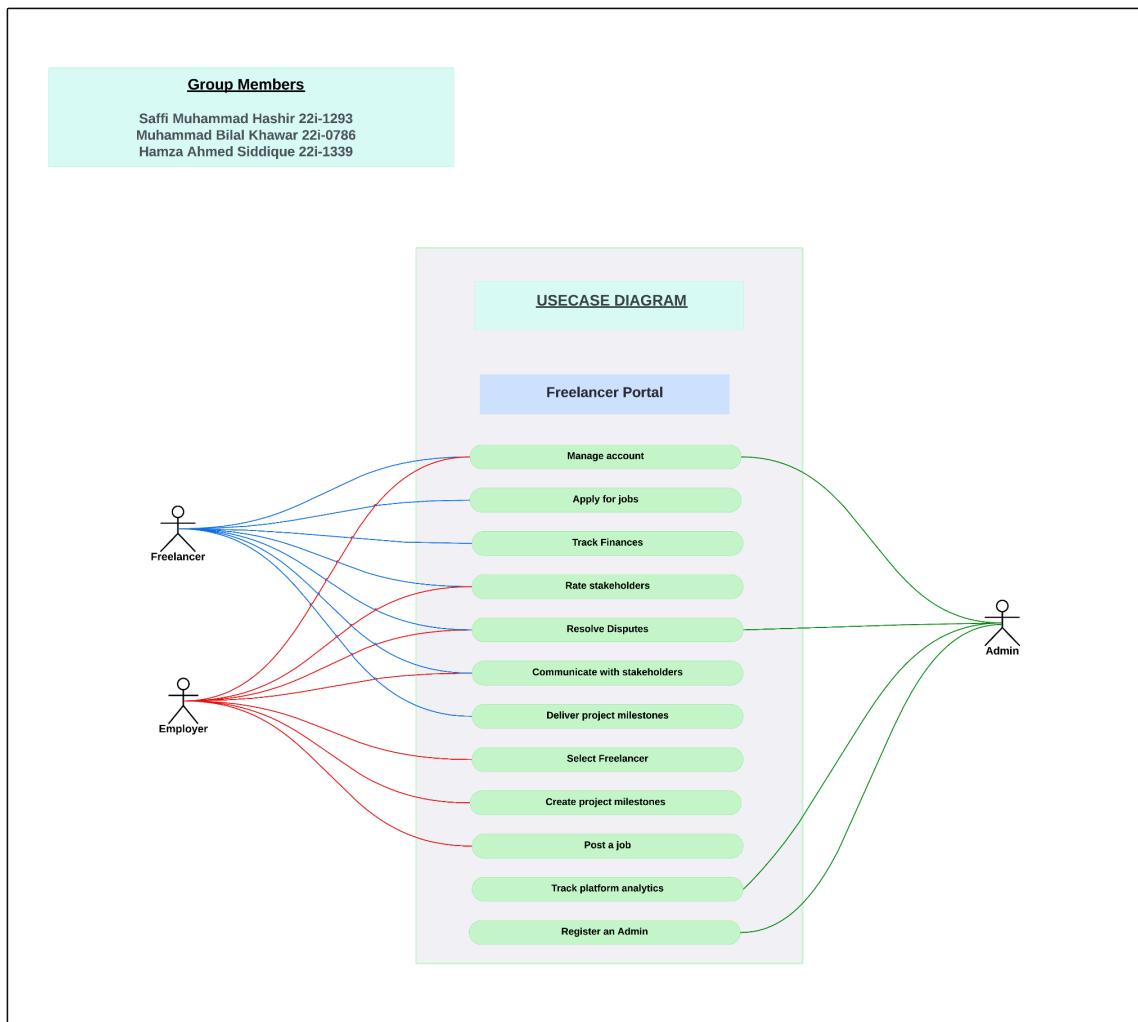
Name	Roll Number	Responsibilities
Muhammad Bilal Khawar	22i-0786	Frontend UI, Jobs Page, Messaging
Saffi M. Hashir	22i-1293	Backend APIs, Database
Hamza Ahmed Siddique	22i-1339	Frontend UI, User System, Trello

## 7.6 Technology Stack

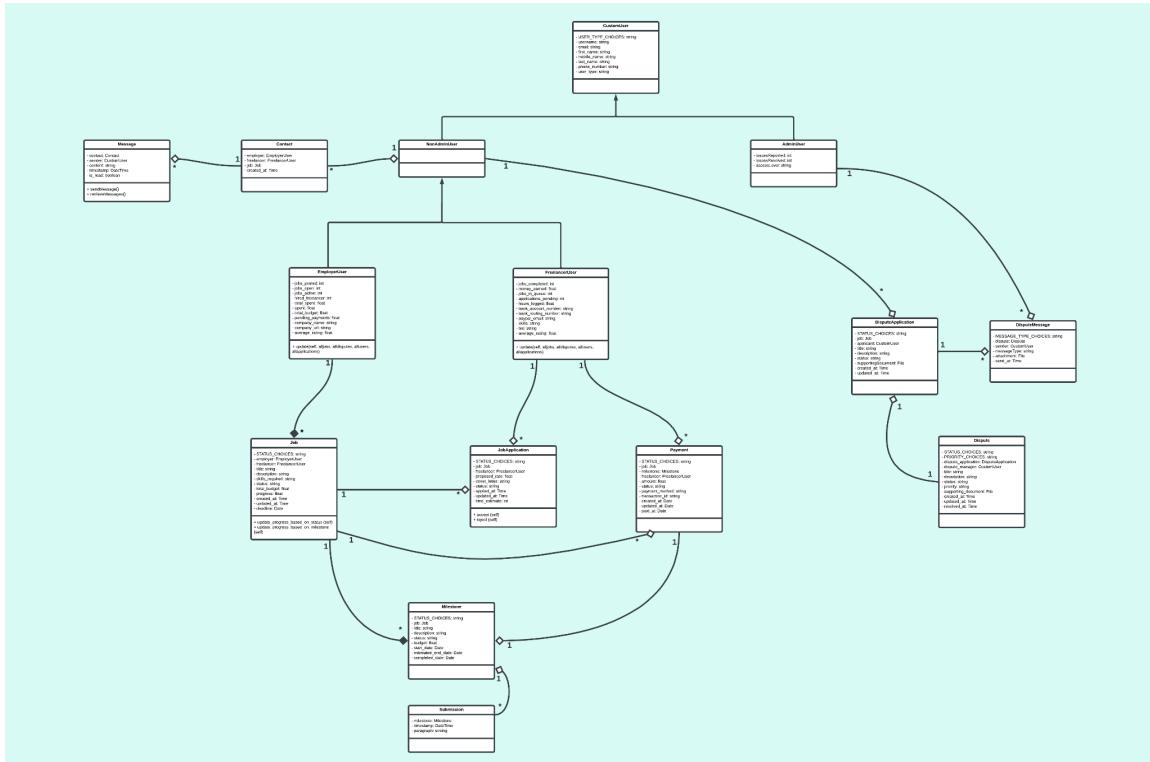
- **Frontend:** HTML, CSS
- **Backend:** Django (Python)
- **Version Control:** Git + GitHub
- **Task Management:** Trello

## 8. System Architecture

### 8.1 Use Case Diagram

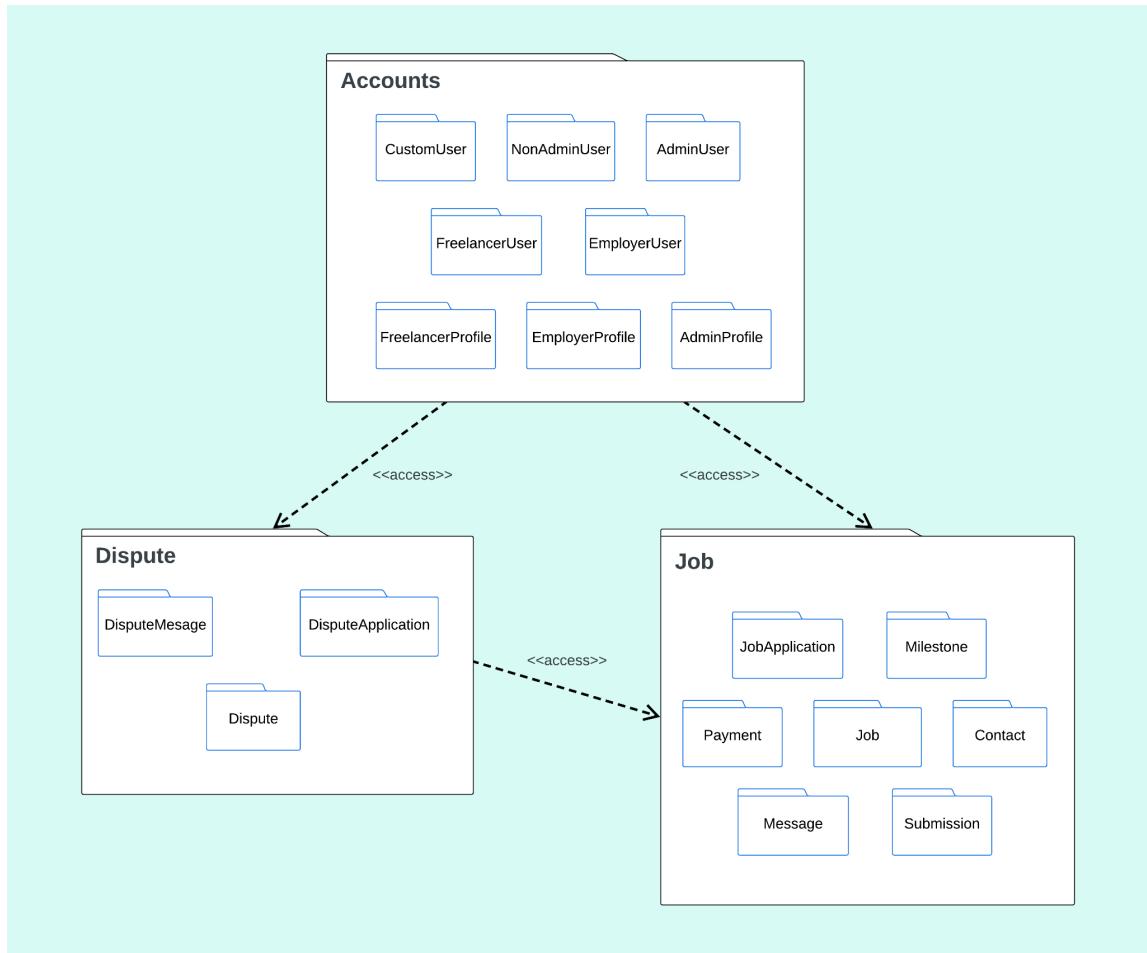


## 8.2 Class Diagram



## 8.3 Package Diagram

Diagram:



**Subsystem Descriptions:**

➤ **Accounts Package**

- **Responsibility:** Manages all user-related functionalities including user creation, role-specific behaviors, and profile management.
- **Key Classes:**
  - **CustomUser, NonAdminUser, AdminUser:** Base and role-specific user models.
  - **FreelancerUser, EmployerUser:** Role-specific user types extending functionality.
  - **FreelancerProfile, EmployerProfile, AdminProfile:** Stores detailed profile information for each user type.

- **Dependencies:** Accessed by both **Job** and **Dispute** packages for user authentication, ownership, and interaction.

#### ➤ Job Package

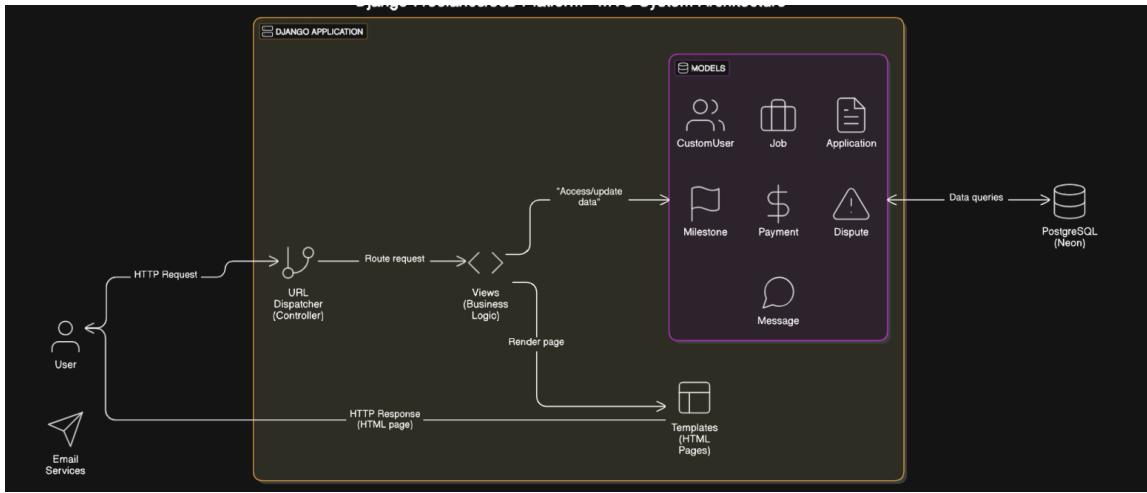
- **Responsibility:** Handles all core job marketplace features including job listings, applications, payments, milestones, communication, and submissions.
- **Key Classes:**
  - **Job:** Represents a freelance job/project.
  - **JobApplication:** Tracks applications by freelancers.
  - **Milestone:** Represents task-based progress checkpoints.
  - **Payment:** Handles milestone or job-level payments.
  - **Message:** Manages communication between employer and freelancer.
  - **Submission:** Allows freelancers to submit deliverables.
  - **Contact:** Handles contact forms or support-related interactions.
- **Dependencies:** Depends on **Accounts** for linking jobs and applications to users.

#### ➤ Dispute Package

- **Responsibility:** Facilitates resolution of conflicts between users (e.g., failed milestones, payment issues).
- **Key Classes:**
  - **Dispute:** Core model representing a raised conflict.
  - **DisputeApplication:** Links a dispute to a specific job application or milestone.
  - **DisputeMessage:** Conversation or evidence exchange related to a dispute.
  - **DisputeManager:** Handles lifecycle and management of disputes.
  - **DisputeResolution:** Stores outcomes or verdicts made by the admin.
- **Dependencies:**
  - Accesses **Accounts** to identify involved users.
  - Accesses **Job** to refer to applications, payments, or milestones under dispute.

## 8.4 Architecture Diagram

Diagram:



### Description of Architectural Styles Used:

The architecture of the system is primarily based on the **Model-View-Controller (MVC)** paradigm, supported by elements of **Layered Architecture** and **Client-Server Architecture**. The following styles are integrated into the system design:

#### ➤ Model-View-Controller (MVC)

The application adopts the MVC architectural pattern, as reflected in the Django framework:

- **Model:** Represents the data and business rules. The system defines various models such as **CustomUser**, **Job**, **Milestone**, **Payment**, **Dispute**, etc., responsible for interacting with the database.
- **View:** Corresponds to Django's template system, which handles the presentation logic and renders dynamic HTML pages for the user.
- **Controller:** Implemented using Django's views and the URL Dispatcher, which route incoming HTTP requests to appropriate view functions containing the business logic.

#### ➤ Layered Architecture

The design also follows a layered (or tiered) structure that separates the system into distinct layers:

- **Presentation Layer:** Handles user interaction through HTML pages.
- **Business Logic Layer:** Manages application logic through view functions.

- **Data Access Layer:** Consists of models that directly interact with the database.

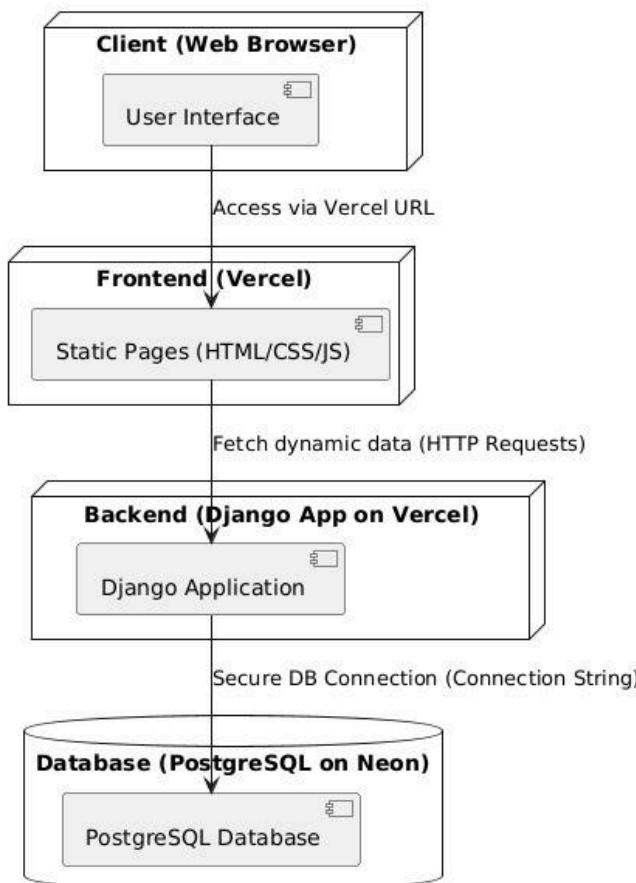
This separation improves modularity, testability, and scalability of the application.

### ➤ Client-Server Architecture

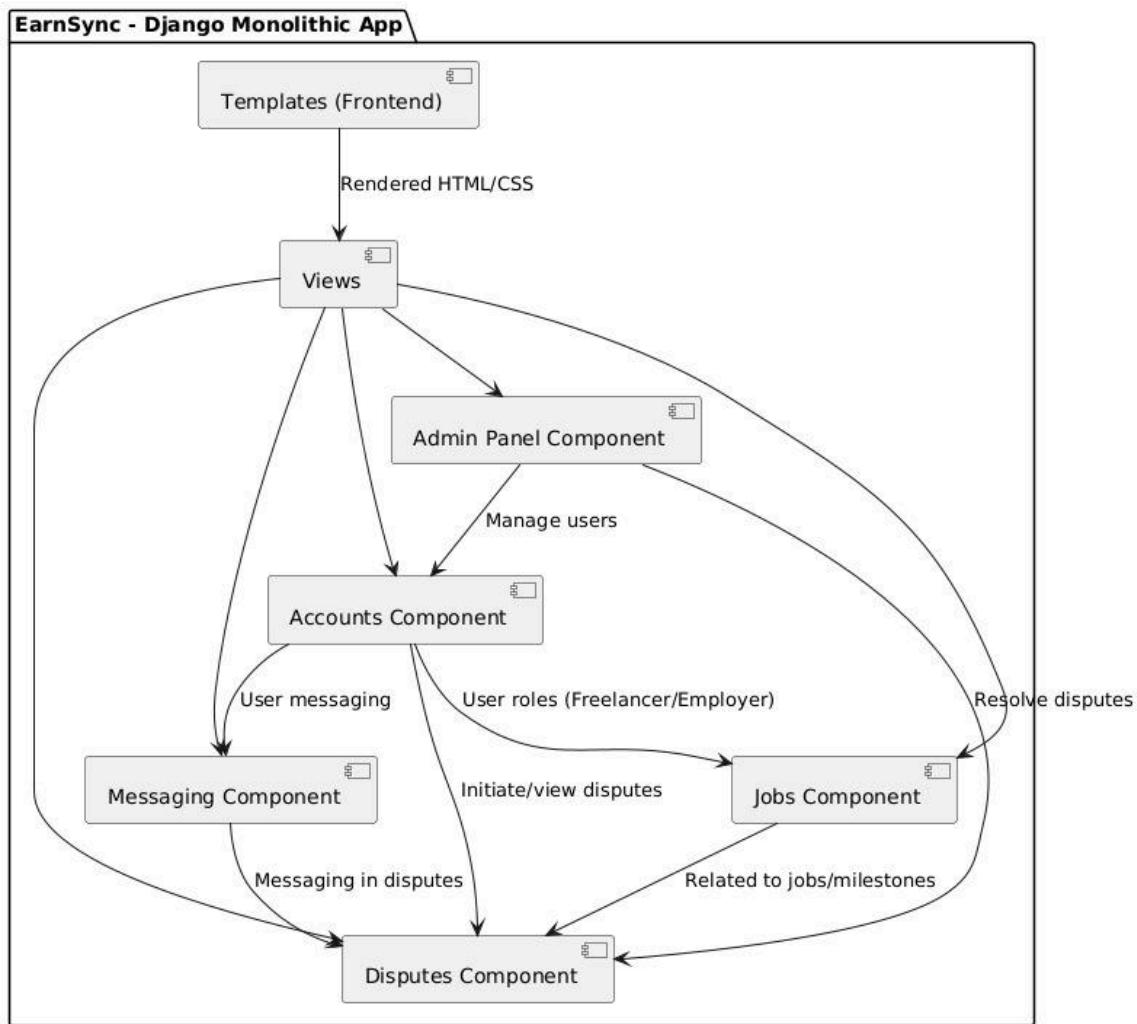
The system clearly adheres to a client-server architecture:

- The **client** (User/Web browser) sends HTTP requests.
- The **server** (Django application) processes the requests, performs the required operations, and returns HTTP responses.

## 8.5 Deployment Diagram



## 8.6 Component Diagram



## **9. Sprint 3 Items**

### a. Projects Page – US01, 09, 10, 14, 15, 16 (Backend)

<i>Description</i>	Backend logic and APIs for posting jobs, applying, listing projects, reviewing applications, and managing statuses.
<i>Priority</i>	High
<i>Assigned to</i>	Hashir

b. Messages Page – US02, 03, 04 (Backend)

<i>Description</i>	Backend services for sending/receiving messages, chat threading, and conversation tracking.
<i>Priority</i>	Medium
<i>Assigned to</i>	Hashir

c. Profile Page – US11, 12, 13 (Backend)

<i>Description</i>	Backend implementation for viewing/editing user profiles for Freelancer, Employer, and Admin.
<i>Priority</i>	Medium
<i>Assigned to</i>	Hashir

d. Payment System – US05, 17 (Frontend)

<i>Description</i>	Frontend for initiating payments, payment confirmation, and viewing transaction history.
<i>Priority</i>	High
<i>Assigned to</i>	Hamza

e. **Payment System – US05, 17 (Backend)**

<i>Description</i>	Backend logic for handling payment processing, gateway integration, and payment validations.
<i>Priority</i>	High
<i>Assigned to</i>	Hamza

f. **Dispute System – US06, 07, 08 (Frontend)**

<i>Description</i>	User interface for raising disputes, communicating during dispute resolution, and viewing resolution history.
<i>Priority</i>	Medium
<i>Assigned to</i>	Bilal

g. **Dispute System – US06, 07, 08 (Backend)**

<i>Description</i>	Backend support for dispute initiation, messaging, management, and resolution workflows.
<i>Priority</i>	Medium
<i>Assigned to</i>	Hamza

**h. View Analytics – US18 (Frontend)**

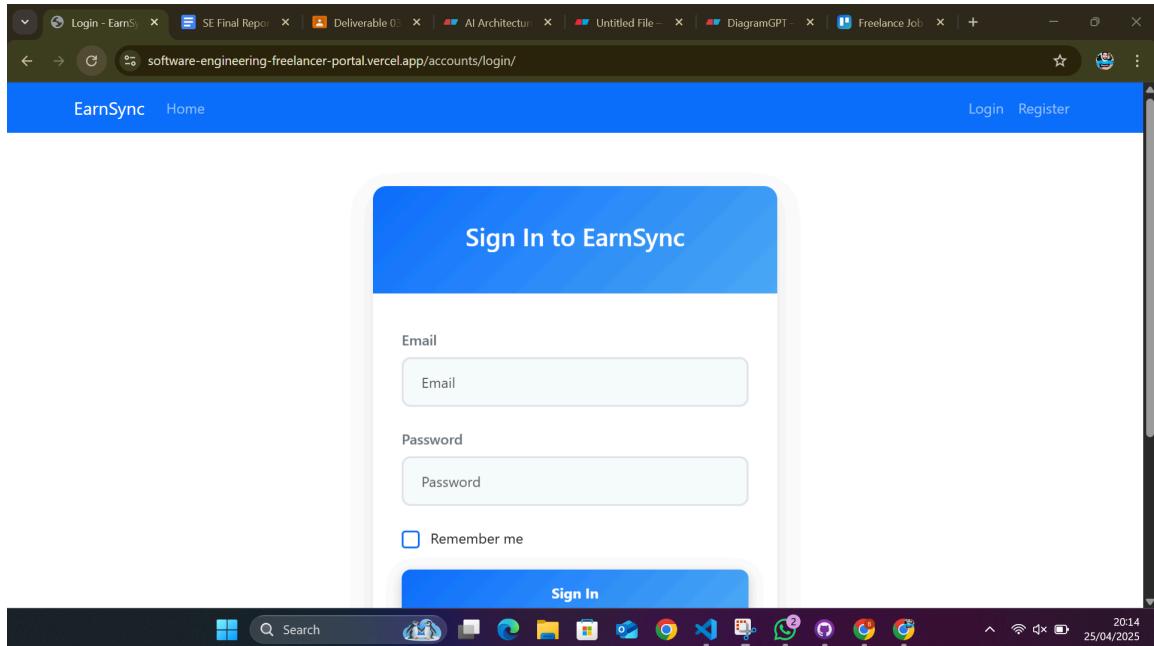
<i>Description</i>	Visual dashboards to show platform activity, user stats, and performance metrics.
<i>Priority</i>	Medium
<i>Assigned to</i>	Bilal

**i. View Analytics – US18 (Backend)**

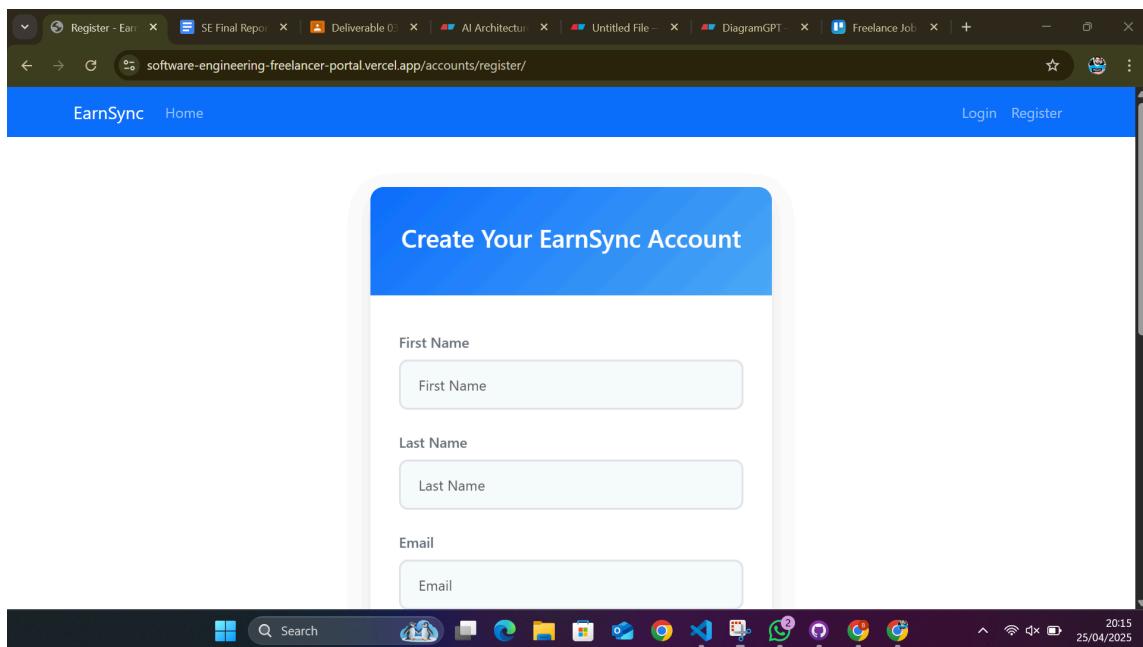
<i>Description</i>	Data aggregation, metrics tracking, and API endpoints for analytics visualizations.
<i>Priority</i>	Medium
<i>Assigned to</i>	Hamza

## 10. Implementation Screenshots

### a. Login



### b. Register



### c. Freelancer's Dashboard

The Freelancer's Dashboard is a web-based application interface. It features a sidebar on the left labeled "EarnSync" with navigation links for Dashboard, Profile, Projects, Messages, and Settings, along with a Logout button. The main content area starts with a welcome message "Welcome back, Muhammad Bilal!" and a subtitle "Track your freelance projects, earnings, and opportunities." Below this is a row of four cards: "Active Projects" (0), "Monthly Earnings" (\$0.00), "Hours Logged" (0.00), and "Proposals Sent" (0). The "Active Projects" section lists columns for Project Name, Client, Deadline, and Progress. The "Available Jobs" section shows two job listings: "Real-Time Cricket Shot Prediction System" and "Freelancer Portal with AI-Powered Job Matching". On the right side, there is a "Quick Actions" panel with buttons for Find New Jobs, Create Invoice, Log Time, and Update Profile. A "Upcoming Deadlines" section is also present. The bottom of the screen shows a Windows taskbar with various pinned icons.

### d. Employer's Dashboard

The Employer's Dashboard is a web-based application interface. It has a similar structure to the Freelancer's Dashboard, with a "EarnSync" sidebar and a main content area. The welcome message is "Welcome back, Muhammad Bilal!" with the subtitle "Manage your projects, freelancers, and job postings." The dashboard includes four cards: "Active Projects" (0), "Total Spent" (\$0.00), "Hired Freelancers" (0), and "Open Positions" (0). The "Active Projects" section lists columns for Project Name, Assigned To, Status, and Progress. The "Recent Applications" section shows a list of recent job applications. The "Quick Actions" panel on the right includes buttons for Post New Job, Hire Freelancer, Invoice, and Manage Contracts. The bottom of the screen shows a Windows taskbar with various pinned icons.

### e. Admin's Dashboard

The screenshot shows the Admin Dashboard for the EarnSync platform. The left sidebar, titled "EarnSync", includes links for Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled "Admin Dashboard" and describes it as a platform overview, user management, and system monitoring. It features three summary cards: "Total Users" (5), "Active Projects" (5), and "Platform Revenue" (\$0). Below these are sections for "Recent User Registrations" and "Quick Actions". The "Recent User Registrations" table lists one user: Saffi Muhammad Hashir, who joined on April 25, 2025, at 2:24 p.m., and is marked as "Active". The "Quick Actions" section contains four buttons: "Add User" (blue), "Manage Roles" (green), "System Settings" (yellow), and "Reports" (teal). A "User Distribution" chart is partially visible at the bottom right.

### f. Profile Page

The screenshot shows the Profile Page for Muhammad Bilal. The left sidebar is identical to the Admin Dashboard. The main content area is titled "Muhammad Bilal's Profile" and provides a quick overview of personal details. It includes fields for First Name (Muhammad Bilal), Middle Name (None), Last Name (Khawar), Email (i220786@nu.edu.pk), and Phone Number (03214943206). Below this is a section titled "Freelancer Profile Option" with fields for Jobs Completed (0), Money Earned (\$: 0.00), and Jobs in Queue (0). At the bottom, there are sections for Applications Pending and Hours Logged. The status bar at the bottom right shows the time as 20:41 and the date as 25/04/2025.

### **g. Projects Page**

The screenshot shows the 'Project Management' section of the EarnSync application. On the left, a sidebar menu includes 'Dashboard', 'Profile', 'Projects' (which is selected and highlighted in blue), 'Messages', and 'Settings'. Below the sidebar is a 'Logout' button. The main content area has a heading 'Project Management' with a sub-instruction 'Want to look at how your on-going projects are progressing or looking for more work?'. There are two buttons: 'Active Projects' (highlighted in blue) and 'Find Projects'. A table titled 'Active Projects' lists five entries:

Project Name	Client	Status	Progress
Freelancer Portal with AI-Powered Job Matching	Chrio	Open	<div style="width: 20%;">20%</div>
AI-Powered Homework Checker	Chrio	Open	<div style="width: 30%;">30%</div>
AI-Powered Code Refactoring Tool	Chrio	In Progress	<div style="width: 70%;">70%</div>
Real-Time Video Captioning System	Chrio	Open	<div style="width: 10%;">10%</div>
Real-Time Cricket Shot Prediction System	Chrio	Open	<div style="width: 5%;">5%</div>

The bottom of the screen shows a Windows taskbar with various pinned icons and the date/time: 20:24 25/04/2025.

### **h. Messages Page**

The screenshot shows the 'Messages' page of the EarnSync application. The sidebar menu is identical to the one in the previous screenshot. The main content area has a 'Chats' section with a list of contacts: 'Saffi Hashir'. The right side shows a detailed chat window for 'Saffi Hashir' with the following messages:

- Saffi Hashir: hello  
2025-04-27 19:10 • Sent
- You: hi  
2025-04-27 19:10
- You: how are u  
2025-04-27 19:11
- Saffi Hashir: i'm fine  
2025-04-27 19:11 • Sent
- Saffi Hashir: what about you?  
2025-04-27 19:11

At the bottom, there is an input field 'Type a message...' and a 'Send' button. The bottom of the screen shows a Windows taskbar with the date/time: 00:11 28/04/2025.

## i. Projects Detail Page

The screenshot shows a browser window displaying a project detail page for an AI-Powered Code Refactoring Tool. The page has a blue header bar with the EarnSync logo. On the left, there's a sidebar with navigation links: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled "AI-Powered Code Refactoring Tool" and shows the employer as "Chrio". A status indicator "In Progress" is visible. Below the title, there's a "Description" section with a detailed text about the tool's purpose, technologies (Python, GPT-based models, AST, Flask/Django), and features. To the right, there are "Budget: \$20000.00" and "Deadline: 25 Jul 2025, 07:37 PM". Under "Skills Required:", there are four tags: Python, GPT-based models, AST (Abstract Syntax Tree), and Flask/Django. A red "Submit a Dispute" button is at the bottom. The browser taskbar at the bottom shows multiple open tabs, and the system tray indicates the date as 25/04/2025.

The screenshot shows a browser window displaying the "Applications" section of the AI-Powered Code Refactoring Tool. The interface is similar to the previous screenshot, with the blue EarnSync header and sidebar. The main content area is titled "Applications". It lists one application from "Saffi Muhammad Hashir" with a rate of "\$20000.00" and a status of "Accepted". The cover letter for this application is displayed in a large text area:

Dear Hiring Manager, I am excited to submit my application for the development of the AI-Powered Code Refactoring Tool. With extensive experience in AI/ML, Python development, and web technologies, I am confident that I can contribute significantly to building a tool that will revolutionize code optimization and refactoring. My background in AI-powered solutions, specifically working with GPT-based models and code analysis using Abstract Syntax Trees (AST), aligns well with the requirements of this project. I have successfully integrated machine learning models into practical applications, optimizing codebases for efficiency and readability. Additionally, my expertise in Flask and Django ensures I can create a robust web interface for seamless interaction with the tool. Key strengths I bring include: Expertise in Python and AI, especially leveraging GPT models for tasks like code analysis and optimization. In-depth experience with AST for parsing and analyzing code. Strong backend development skills in Flask and Django for web interface development. Proven ability to integrate with GitHub APIs, enabling code refactoring within repositories. I am excited about the potential impact of this project and look forward to the opportunity to contribute to its success. Thank you for considering my application. I would welcome the chance to further discuss how my skills and experience can help achieve your goals for this tool.

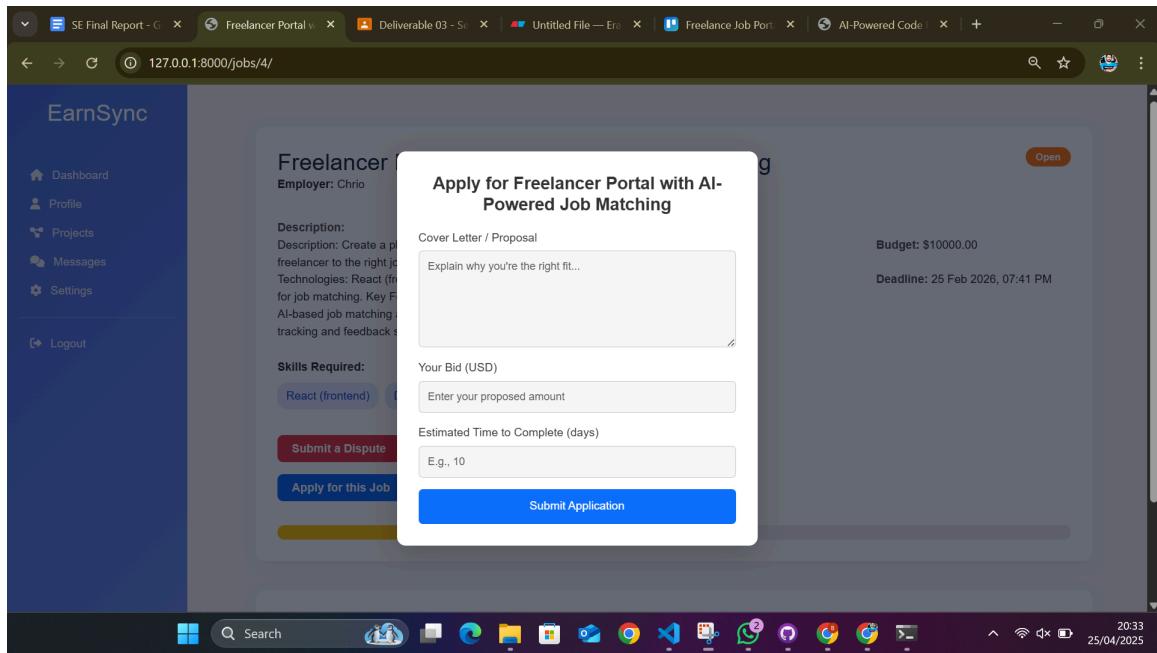
A blue "View" button is located to the right of the cover letter. The browser taskbar and system tray are visible at the bottom, showing the date as 25/04/2025.

The screenshot shows the EarnSync application interface. On the left, a blue sidebar menu includes 'Dashboard', 'Profile', 'Projects', 'Messages', 'Settings', and a 'Logout' button. The main content area has two sections: 'Milestones' and 'Payments'.  
**Milestones:**  
A table with columns: Title, Status, Budget, Start, Due, Completed, and Actions. Data rows:  
1. Requirement Analysis & Initial Prototype: Completed, \$400.00, 09 May 2025, April 25, 2025, 2:59 p.m., View  
2. Develop Core Refactoring Logic: In Progress, \$600.00, 13 Jun 2025, -, View  
3. Web Interface & Deployment: In Progress, \$700.00, 20 Jun 2025, -, View  
**Payments:**  
A table with columns: Milestone, Freelancer, Amount, Status, Transaction ID, Paid At, and Actions. Data row:  
Requirement Analysis & Initial Prototype: Saffi Muhammad Hashir, \$400.00, Completed, TXN-20250425145904-4, 25 Apr 2025, View

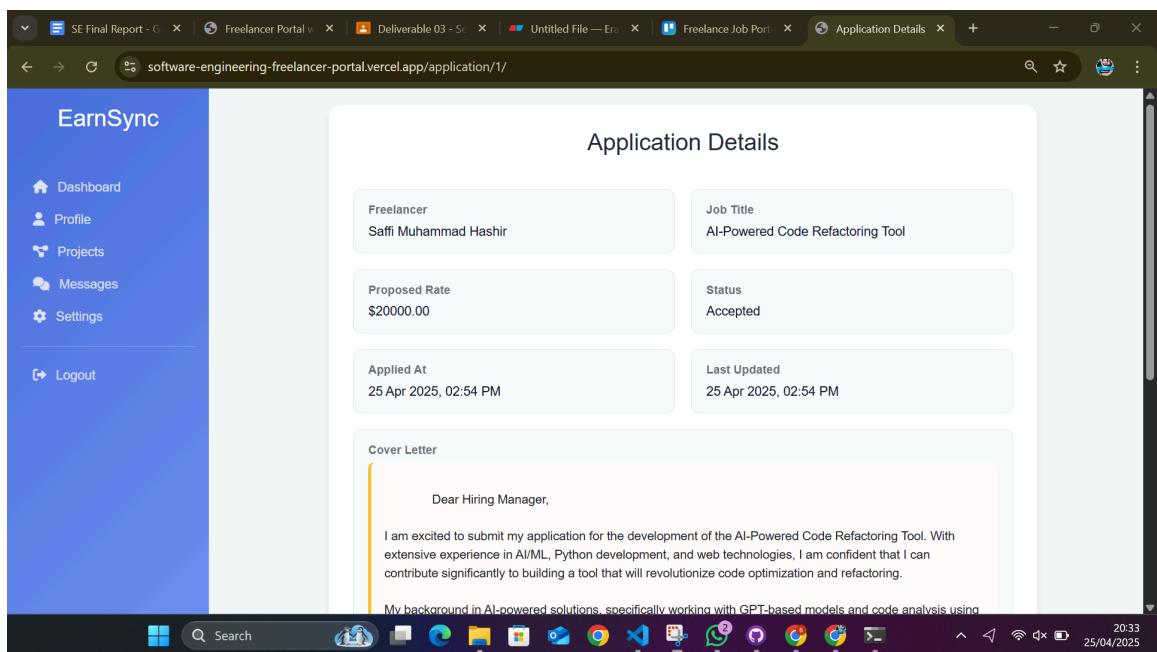
#### j. Submit a Dispute

The screenshot shows the 'Submit a Dispute' form within the EarnSync application. The left sidebar is identical to the previous screenshot. The main form has a title 'Submit a Dispute' and two input fields: 'Dispute Title:' and 'Description:'. A large blue 'Submit Dispute' button is at the bottom.

## k. Apply for Job



## l. Application Detail



### m. Milestone Detail

The screenshot shows the 'Milestone Details' page within the 'EarnSync' application. The left sidebar contains navigation links: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Milestone Details' and displays the following information:

- Title:** Develop Core Refactoring Logic
- Job:** AI-Powered Code Refactoring Tool
- Description:** Sub-budgets:
  - AI model integration (GPT-based model): \$500
  - Parsing tools (e.g., AST): \$100
- Budget:** \$600.00
- Status:** In Progress
- Start Date:**
- Expected End Date:** 13 Jun 2025

The browser address bar shows the URL: software-engineering-freelancer-portal.vercel.app/milestone/5. The system status bar at the bottom right indicates the time as 20:33 and the date as 25/04/2025.

### n. Payment Detail

The screenshot shows the 'Payment Details' page within the 'EarnSync' application. The left sidebar contains navigation links: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Payment Details' and displays the following information:

- Milestone:** Requirement Analysis & Initial Prototype
- Freelancer:** Saffi Muhammad Hashir
- Amount:** \$400.00
- Status:** Completed
- Transaction ID:** TXN-20250425145904-4
- Payment Method:** manual
- Created At:** 25 Apr 2025, 02:59 PM
- Updated At:** 25 Apr 2025, 02:59 PM

The browser address bar shows the URL: software-engineering-freelancer-portal.vercel.app/payment/1. The system status bar at the bottom right indicates the time as 20:34 and the date as 25/04/2025.

## **o. Add Project**

The screenshot shows a browser window with the URL [software-engineering-freelancer-portal.vercel.app/addProject/](https://software-engineering-freelancer-portal.vercel.app/addProject/). The page has a blue sidebar on the left labeled 'EarnSync' with options: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Add a New Project' and contains fields for Project Title, Project Description, Required Skills (comma-separated), Total Budget (\$), and Deadline. Below the form is a Windows taskbar with various pinned icons.

## **p. Manage Contracts**

The screenshot shows a browser window with the URL [software-engineering-freelancer-portal.vercel.app/contracts](https://software-engineering-freelancer-portal.vercel.app/contracts). The page has a blue sidebar on the left labeled 'EarnSync' with options: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Contracts' and lists three projects: 'Website Redesign Project' (Client: Acme Corp, Budget: \$2000, Duration: 30 days, status: Pending), 'Mobile App Development' (Client: Beta Tech, Budget: \$5000, Duration: 60 days, status: Accepted), and 'UI/UX Overhaul'. Below the contracts is a Windows taskbar with various pinned icons.

**q. Invoices**

The screenshot shows the 'Payments employer' section of the EarnSync platform. The left sidebar has a blue header 'EarnSync' and navigation links: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Payments employer' with the subtitle 'Payments Made Easy, So Y'. It features a heading 'Milestone Payments' and a table with the following data:

Project Name	Client	Payment Details	Payment Status
Real-Time Cricket Shot Prediction System	Data Collection and Preprocessing	\$700.00	<span style="background-color: yellow; border: 1px solid black; padding: 2px;">Hold</span>
Real-Time Cricket Shot Prediction System	Model Development and Training	\$500.00	<span style="background-color: yellow; border: 1px solid black; padding: 2px;">Hold</span>
Real-Time Cricket Shot Prediction System	Integration and Real-Time Prediction	\$1000.00	<span style="background-color: yellow; border: 1px solid black; padding: 2px;">Hold</span>
AI-Powered Homework Checker	Develop Textual Grading System (NLP)	\$500.00	<span style="background-color: yellow; border: 1px solid black; padding: 2px;">Hold</span>
AI-Powered Homework Checker	Develop Code Grading System	\$400.00	<span style="background-color: yellow; border: 1px solid black; padding: 2px;">Hold</span>
AI-Powered Homework Checker	Build Web Interface & User Reports	\$800.00	<span style="background-color: yellow; border: 1px solid black; padding: 2px;">Hold</span>

The bottom status bar shows system icons and the date/time: 20:40 25/04/2025.

**r. Resolve Dispute**

The screenshot shows the 'Resolve Dispute' section of the EarnSync platform. The left sidebar has a blue header 'EarnSync' and navigation links: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Disputes for Job: AI-Powered Code Refactoring Tool' and lists two disputes:

- Dispute ID: 1**  
Created At: April 25, 2025, 3:06 p.m.  
Description: asdfadsfdfs  
Buttons: Accept Conflict, View Messages, Resolve Dispute (Pending)
- Dispute ID: 2**  
Created At: April 25, 2025, 3:29 p.m.  
Description: yujjhfgshaassfasdfasdfsrrf  
Buttons: Accept Conflict, View Messages, Resolve Dispute (Pending)

The bottom status bar shows system icons and the date/time: 22:33 25/04/2025.

**s. Add a User**

The screenshot shows a browser window with the URL [software-engineering-freelancer-portal.vercel.app/addUser/](https://software-engineering-freelancer-portal.vercel.app/addUser/). The page has a blue sidebar on the left labeled 'EarnSync' with options: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Add a New User' and contains fields for First Name, Last Name, Email, Password, and Phone Number. Below these fields is a 'User Type' dropdown menu. The bottom of the screen shows a Windows taskbar with various icons.

**t. Reports**

The screenshot shows a browser window with the URL [software-engineering-freelancer-portal.vercel.app/reports](https://software-engineering-freelancer-portal.vercel.app/reports). The page has a blue sidebar on the left labeled 'EarnSync' with options: Dashboard, Profile, Projects, Messages, Settings, and Logout. The main content area is titled 'Reports' and includes a sub-header: 'Monitor and analyze system-wide transactions, user activity, and financial reports.' It features a 'Date Range' selector with two input fields and a 'Apply' button. Below this are three summary boxes: 'Total Payments \$12,500', 'Active Users 342', and 'Pending Payments \$3,200'. At the bottom is a table with columns: Transaction ID, Sender, Reciever, Amount, Date, and Status. One row is visible: #TX001, John Doe, Jane Doe, \$200, 2025-04-16, Completed. The bottom of the screen shows a Windows taskbar with various icons.

Transaction ID	Sender	Reciever	Amount	Date	Status
#TX001	John Doe	Jane Doe	\$200	2025-04-16	Completed

## u. View User Details

The screenshot shows the 'User Details' page of the EarnSync application. The left sidebar has a blue header 'EarnSync' and links for Dashboard, Profile, Projects, Messages, and Settings, with a Logout button. The main content area has a title 'User Details' under 'Personal Details'. It displays four fields: Full Name (Saffi Muhammad Hashir), Email (saffimhashir123@gmail.com), Phone (03126247183), and User Type (Freelancer). Below this is a section titled 'Performance' with two boxes: Total Projects (10) and Completed Projects (8). The bottom status bar shows the date 25/04/2025 and time 20:44.

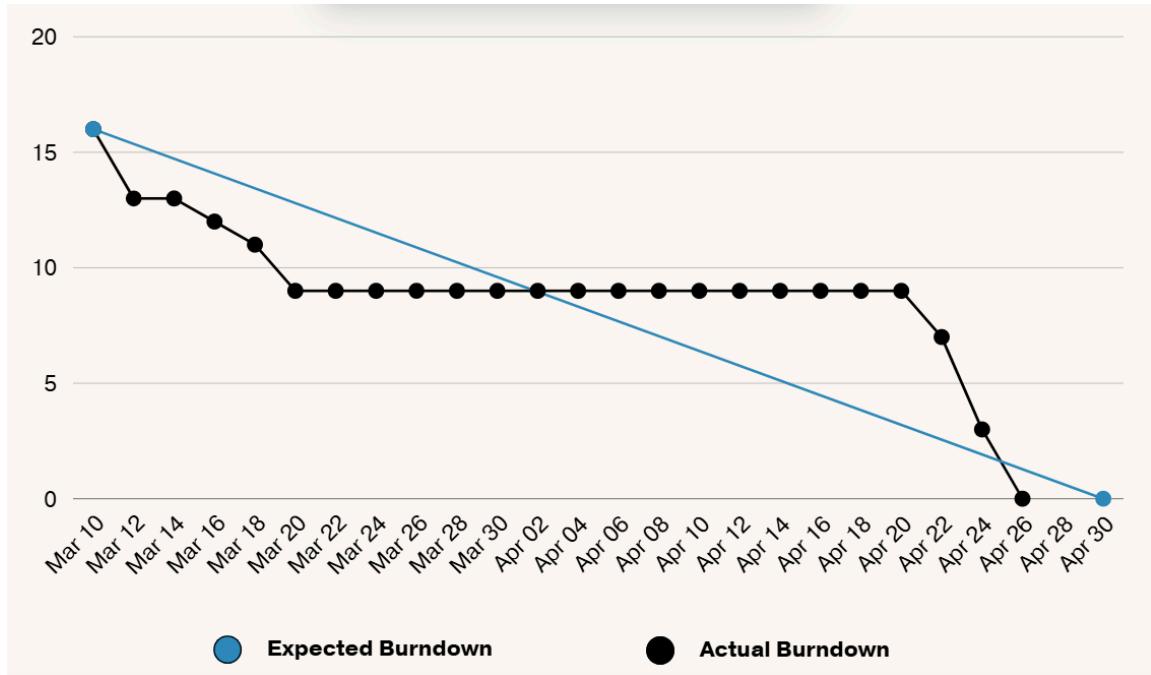
The screenshot shows the 'User Details' page of the EarnSync application. The left sidebar has a blue header 'EarnSync' and links for Dashboard, Profile, Projects, Messages, and Settings, with a Logout button. The main content area has a title 'Performance' with two boxes: Total Projects (10) and Completed Projects (8). Below this is a section titled 'Projects' with a table:

Title	Status	Budget	Start Date	End Date
CRM Dashboard	Completed	\$5,000	Jun 01, 2022	Aug 15, 2022
E-commerce Site	Ongoing	\$7,000	Jan 10, 2023	Apr 10, 2023
Portfolio Website	Completed	\$1,500	Nov 20, 2021	Dec 05, 2021

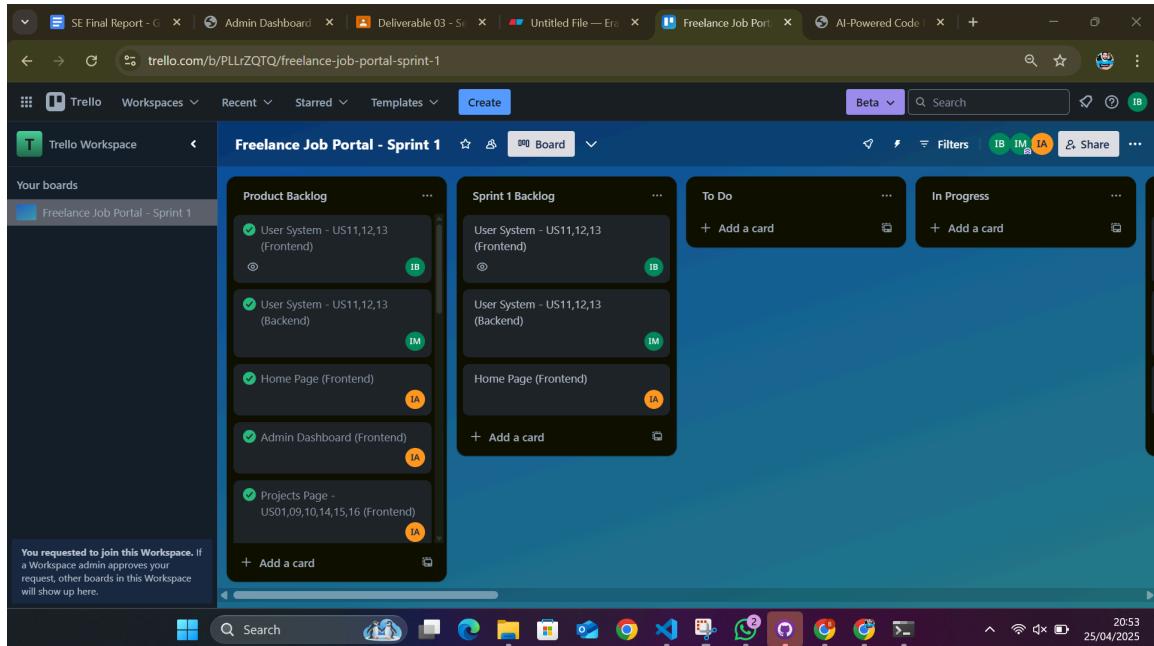
Below the projects is a section titled 'Skills'.

The bottom status bar shows the date 25/04/2025 and time 20:44.

## 11. Burndown chart



## 12. Trello Board Screenshots



SE Final Report - G | Admin Dashboard | Deliverable 03 - Se | Untitled File — Era | Freelance Job Port... | AI-Powered Code

trello.com/b/PLLrZQTQ/freelance-job-portal-sprint-1

Trello Workspaces | Recent | Starred | Templates | Create | Beta | Search | Filters | Share | ...

Freelance Job Portal - Sprint 1 | Board | Filters | Share | ...

Your boards | Freelance Job Portal - Sprint 1

Sprint 2 Backlog | To Do | In Progress | Completed

- Admin Dashboard (Frontend) | IA
- Projects Page - US01,09,10,14,15,16 (Frontend) | IA
- Messages Page - US02,03,04 (Frontend) | IB
- Profile Page - US11,12,13 (Frontend) | IB

+ Add a card

You requested to join this Workspace. If a Workspace admin approves your request, other boards in this Workspace will show up here.

20:53 25/04/2025

This screenshot shows the Trello board for Sprint 2 Backlog. It has four columns: To Do, In Progress, and Completed, each with a 'Add a card' button. The 'To Do' column contains four cards: Admin Dashboard (Frontend), Projects Page - US01,09,10,14,15,16 (Frontend), Messages Page - US02,03,04 (Frontend), and Profile Page - US11,12,13 (Frontend). Each card has an 'IA' or 'IB' priority indicator. A sidebar message indicates that other boards in the workspace will appear once approved.

SE Final Report - G | Admin Dashboard | Deliverable 03 - Se | Untitled File — Era | Freelance Job Port... | AI-Powered Code

trello.com/b/PLLrZQTQ/freelance-job-portal-sprint-1

Trello Workspaces | Recent | Starred | Templates | Create | Beta | Search | Filters | Share | ...

Freelance Job Portal - Sprint 1 | Board | Filters | Share | ...

Your boards | Freelance Job Portal - Sprint 1

Sprint 3 Backlog | To Do | In Progress | Completed

- (Backend) | IA
- Dispute System - US06,07,08 (Frontend) | IB
- Dispute System - US06,07,08 (Backend) | IA
- View Analytics - US18 (Frontend) | IB
- View Analytics - US18 (Backend) | IA

+ Add a card

You requested to join this Workspace. If a Workspace admin approves your request, other boards in this Workspace will show up here.

20:54 25/04/2025

This screenshot shows the Trello board for Sprint 3 Backlog. It has four columns: To Do, In Progress, and Completed, each with a 'Add a card' button. The 'To Do' column contains five cards: (Backend), Dispute System - US06,07,08 (Frontend), Dispute System - US06,07,08 (Backend), View Analytics - US18 (Frontend), and View Analytics - US18 (Backend). Each card has an 'IA' or 'IB' priority indicator. A sidebar message indicates that other boards in the workspace will appear once approved.

## 13. TestCases - Black Box

### 13.1 Equivalence Class Partitioning

Test Case ID	Feature	Test Description	Input	Expected Output	Status
TC01	User Registration	Valid registration input for Employer	<b>Email:</b> <a href="mailto:bilal@gmail.com">bilal@gmail.com</a> <b>Password:</b> test123 <b>Role:</b> Employer	Registration successful	Pass
TC02	User Registration	Invalid email format	<b>Email:</b> bilal@.com <b>Password:</b> test123 <b>Role:</b> Employer	Show error: Invalid email	Pass
TC03	Login	Valid credentials	<b>Email:</b> bilal@gmail.com <b>Password:</b> test123	Redirect to dashboard	Pass
TC04	Login	Invalid password	<b>Email:</b> <a href="mailto:bilal@gmail.com">bilal@gmail.com</a> <b>Password:</b> xyz	Show error: Incorrect password	Pass
TC05	Post Job	Empty title (invalid input class)	<b>Title:</b> ""	Show error: Title required	Pass
TC06	Apply to Job	Valid freelancer applying to open job	<b>Job ID:</b> 2 <b>User:</b> bilal	Application submitted	Pass
TC07	Invalid Message Sending	Empty message text	<b>Message:</b> ""	Show error: Cannot send empty message	Pass
TC08	Valid Message Sending	Proper text message	<b>Message:</b> "Hello, I'm interested"	Message appears in chat thread	Pass

TC09	Dispute Creation	Valid milestone dispute request	<b>Dispute Title:</b> Delay <b>Description:</b> "Work delay"	Dispute submitted	Pass
TC10	Profile Update	Invalid phone number format	<b>Phone:</b> 123abc	Show error: Please match the requested format.	Pass

### 13.2 Boundary Value Analysis

Test Case ID	Feature	Test Description	Input	Expected Output	Status
TC11	Password Validation	Minimum length (boundary)	<b>Password:</b> 8 chars	Accepted	Pass
TC12	Password Validation	Below minimum length	<b>Password:</b> 5 chars	Error: Too short	Pass
TC13	Job Title Length	Maximum allowed characters (boundary)	<b>Title:</b> 255 chars	Accepted	Pass
TC14	Job Title Length	Above maximum limit	<b>Title:</b> 300 chars	Error: Title too long	Pass
TC15	Budget Input	Minimum budget boundary	<b>Budget:</b> 0	Error: Budget must be positive	Pass
TC16	Budget Input	Valid budget just above boundary	<b>Budget:</b> 1	Accepted	Pass
TC17	Description Length	Maximum character limit	<b>Description:</b> 1000 chars	Accepted	Pass

TC18	Description Length	Exceeding max character limit	<b>Description:</b> 1001 chars	Error: Description too long	Pass
TC19	Milestone Budget	Minimum budget boundary	<b>Budget:</b> 1	Accepted	Pass
TC20	Milestone Budget	Valid budget just above boundary	<b>Budget:</b> 0	Error: Invalid rating	Pass

## 14. TestCases - White Box

White box testing was carried out using the pytest-django framework to ensure core logic and form validations of the EarnSync platform functioned correctly. The coverage package was used to measure how much of the internal codebase (models, views, and forms) was executed during testing.

### 14.1 Tools Used

- **pytest-django** to run the unit tests
- **coverage** to generate coverage report

### 14.2 Features Tested

Module	Tests Applied
<b>Authentication</b>	Registration, login, valid/invalid credentials, redirect to dashboards
<b>Forms</b>	CustomUserChangeForm – Valid inputs, invalid email, missing phone number
<b>Validation</b>	Password length enforcement, email format, required fields
<b>Views Permissions</b>	Access control (e.g., forbidden page access for wrong role)
<b>Dashboard Views</b>	Admin, Employer, Freelancer dashboard rendering

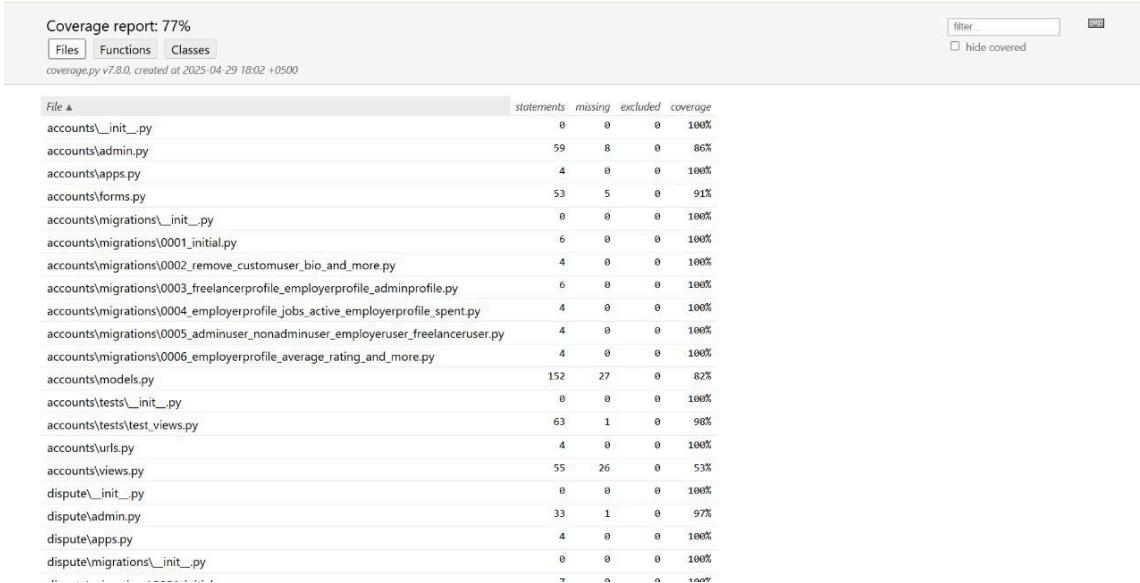
<b>Job &amp; Milestones</b>	Add Job, Job Details, Submit Milestone, Accept Milestone (valid and forbidden)
<b>Dispute Handling</b>	Submit Dispute, Add Dispute, Resolve Dispute
<b>Application Process</b>	Job Application Submission, Hire Freelancer (with and without permission)

## 14.3 Code Coverage Summary

- Statement Coverage: **77%**
- Class Coverage: **77%**
- Function Coverage: **77%**

## 14.4 Results

- A detailed HTML report was generated using coverage:



- Unit testing was done using pytest-django:

```
C:\WINDOWS\system32\cmd. x + v
Test: Accept Milestone Forbidden (Wrong User) - Freelancer correctly blocked from accepting milestone.
.Test: Add Dispute View - Running...
Test: Add Dispute View - Add dispute form loaded successfully.
.Test: Add Job View (GET) - Running...
Test: Add Job View (GET) - Add Job page loaded successfully.
.Test: Admin Dashboard View - Running...
Test: Admin Dashboard View - Admin dashboard displayed.
.Test: Employer Dashboard View - Running...
Test: Employer Dashboard View - Employer dashboard displayed.
.Test: Freelancer Dashboard View - Running...
Test: Freelancer Dashboard View - Freelancer dashboard displayed.
.Test: Hire Freelancer Forbidden - Freelancer access to hire freelancer page correctly forbidden.
.Test: Hire Freelancer View - Running...
Test: Hire Freelancer View - Hire freelancer page loaded.
.Test: Job Detail View - Running...
Test: Job Detail View - Job details displayed successfully.
.Test: Milestone Submission - Running...
Test: Milestone Submission - Milestone submitted successfully.
.Test: Milestone Submission (GET) - Running...
Test: Milestone Submission (GET) - Milestone submission form opened successfully.
.Test: Resolve Dispute - Running...
Test: Resolve Dispute - Dispute resolved successfully.
.Test: Submit Dispute - Running...
Test: Submit Dispute - Dispute submitted successfully.
.Test: Submit Job Application - Running...
Test: Submit Job Application - Application submitted successfully.
.

Ran 17 tests in 11.557s
OK
Destroying test database for alias 'default'...
Black Box Testing Done
Press any key to continue . . . |
```

## **14.5 What Is Covered vs Not Covered**

- **Covered Well:**
  - User registration, login, redirects
  - Dashboard access for all user roles
  - Form validation for custom user model
  - Milestone submission & completion workflows
  - Dispute creation and resolution
  - Role-based access restrictions
  - Job posting, application & hiring
- **Not Covered:**
  - Third-party Django auth internals
  - External payment gateway integrations
  - Frontend UI logic and JavaScript components
  - Asynchronous tasks (email notifications, reminders)

**Note:** These areas were not tested because they rely on built-in Django modules, frontend code, or background tasks not covered by backend unit tests.

## 15. Work Division between Group Members

The group followed Agile methodology with weekly sprints. Tasks were divided at the start of each sprint during planning meetings. Each member handled full-stack or focused modules, depending on expertise.

Member	Modules/Tasks Worked On
Hamza	<b>Frontend:</b> Home Page, Projects Page, Messages Page, Payment System (Frontend), Analytics UI <b>Backend:</b> Payment System, Dispute System, Analytics Backend
Hashir	<b>Backend:</b> User System (Authentication, Registration APIs), Admin Dashboard, Projects Page, Messages, Profile Management <b>Database:</b> Handled all DB related issues and also dealt with deployment
Bilal	<b>Frontend:</b> User System UI (Login, Signup), Profile Page UI, Dispute System (Frontend), Analytics Page <b>Others:</b> Documentation and Reports

## 16. Lessons Learnt by Group

- **Collaborative Development:** Working as a team helped us understand the importance of version control (Git), effective communication, and collaboration tools like Trello/Slack.
- **Agile Practice:** Implementing sprints and tracking backlogs gave us real experience in Agile software development and iterative delivery.
- **Full Stack Exposure:** We gained practical experience working across frontend (HTML, CSS, JS) and backend (Django).
- **Django Framework:** Deep understanding of Django's MVT pattern, authentication, form handling, and model relationships.
- **Testing and Debugging:** Writing black-box test cases and fixing integration bugs improved our understanding of system validation and robustness.

- **Deployment & Architecture:** Understanding of MVC, client-server architecture, deployment diagram design, and system scalability.
- **Problem Solving:** Dispute resolution, milestone payments, and analytics modules taught us how to tackle real-world project management features.

## 17. Links

- Github:  
<https://github.com/saffimuhammadhashir/Software-Engineering-Freelancer-Portal>
- Trello:  
<https://trello.com/b/PLLrZQTQ/freelance-job-portal-sprint-1>