

CS3009 – Software Engineering

Deliverable 03

Prepared by

Team New Balance

Muhammad Raza Khan 22i-1234

Hassan Afzal 22i-0918

Asjad-Ullah 22i-1227

Date: 27-04-2025

A- WARE SOFTPROJECT PLAN

1. Introduction

New Balance is a freelance job portal designed to connect freelancers and clients on a single platform. It allows clients to post job opportunities and freelancers to bid on them. The system manages registration, profile management, job postings, bidding, communication, real-time notifications, and contract completion. The objective is to create a seamless and secure environment for freelance collaborations.

2. Objectives

- Provide a secure platform for user registration and login.
- Allow clients to post jobs and freelancers to search and apply for jobs.
- Facilitate effective communication between clients and freelancers.
- Enable a rating and review system for quality control.
- Track work history, earnings, and maintain real-time updates through notifications.

3. Work Breakdown Structure (WBS)

1. Project Initiation

1.1 Requirements Gathering

Identify functional and non-functional requirements by analyzing freelance marketplaces like Upwork and Fiverr.

1.2 Use Case Analysis

Define and document use cases to ensure the system meets the needs of freelancers, clients, and admins.

1.3 Technology Stack Finalization

Choose frontend, backend, database, and hosting solutions (e.g., ReactJS, NodeJS, MongoDB, AWS).

1.4 Initial Wireframes

Design basic sketches for key pages like registration, login, dashboard, job listings, and chat.

2. Frontend Development

2.1 User Registration and Authentication UI

Design and implement user-friendly forms for registration (new users) and login (existing users), including validation messages and error handling.

2.2 Freelancer and Client Dashboard UI

Develop separate dashboard interfaces for freelancers (showing job opportunities, earnings) and clients (managing posted jobs and bids received).

2.3 Job Posting and Search Pages

Create pages where clients can post jobs with detailed information and where freelancers can search, filter, and view job listings easily.

2.4 Bid Submission and Proposal Pages

Build interactive pages allowing freelancers to submit bids and detailed proposals for jobs, and clients to review received proposals.

2.5 Profile Management Pages

Design profile editing pages where users can update personal details, skills, portfolio, company information, and upload profile pictures.

2.6 Messaging and Notification UI

Implement an internal chat system and a notification panel to enable real-time communication between freelancers and clients.

3. Backend Development

3.1 Database Design (Users, Jobs, Bids, Messages)

Design normalized MongoDB schemas for storing user information, job posts, bids, messages, and notifications.

3.2 API Development for User Authentication

Develop REST APIs to manage user registration, login/logout, password reset, and session management securely.

3.3 API Development for Job Posting and Search

Create APIs to allow clients to post jobs and freelancers to search or filter jobs based on criteria.

3.4 API Development for Bidding System

Develop APIs to handle bid submission, proposal management, acceptance/rejection, and notifications.

3.5 API for Messaging and Real-time Notifications

Implement WebSocket-based real-time messaging APIs and notification APIs for immediate user updates.

3.6 API for Work History and Feedback

Create APIs to store and retrieve freelancers' completed jobs, feedback, and client ratings.

3.7 Admin Panel (for system management)

Develop an admin interface to manage users, jobs, disputes, and platform activities.

4. Testing

4.1 Unit Testing

Test individual components (React components, NodeJS APIs) for correctness.

4.2 Integration Testing

Ensure APIs integrate smoothly with the frontend interfaces and data is fetched and displayed properly.

4.3 User Acceptance Testing

Conduct testing with a sample group of users to verify the system meets all expectations and usability standards.

5. Maintenance

5.1 Bug Fixing

Regularly fix bugs reported by users or detected by internal testing.

5.2 Updates and Feature Enhancements

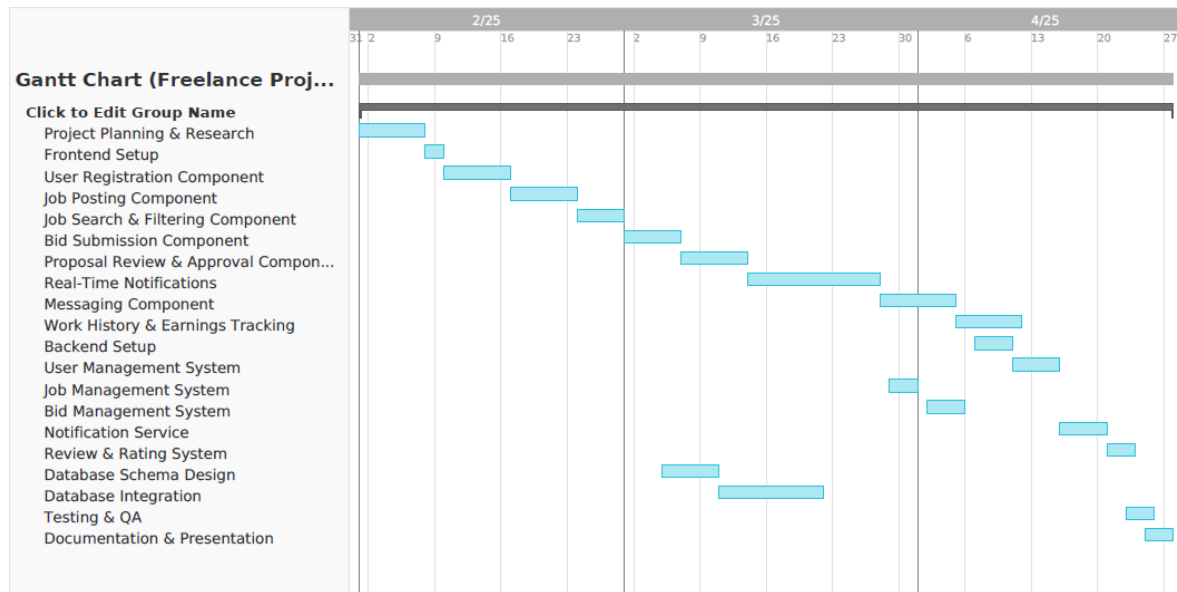
Add new features based on user feedback and evolving market needs.

5.3 System Monitoring

Monitor system performance, server uptime, database usage, and address any technical issues proactively.

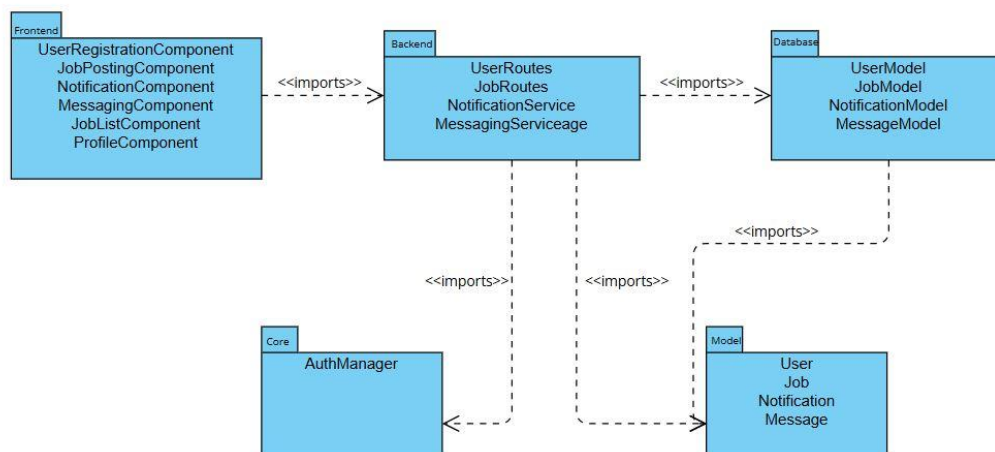
4. Timeline (Gantchart)

teamgantt
Created with Free Edition

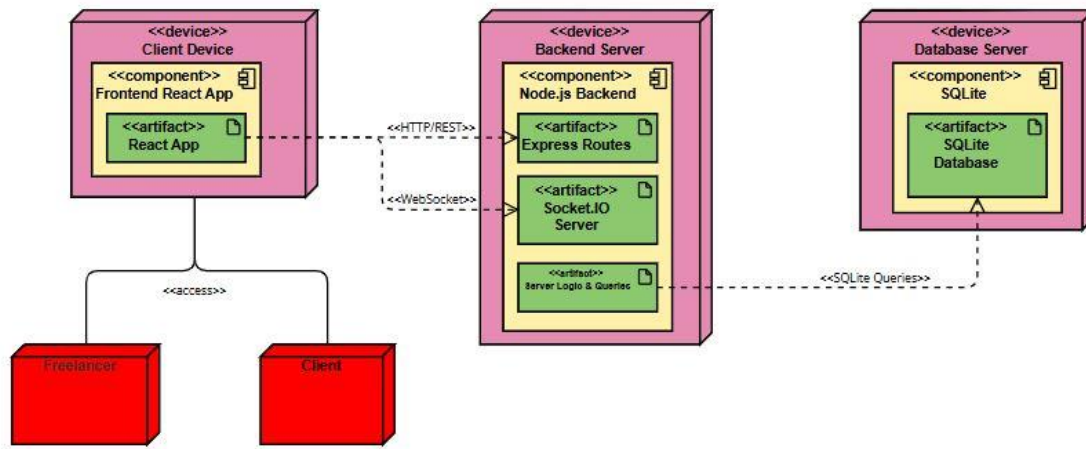


B- SYSTEM ARCHITECTURE

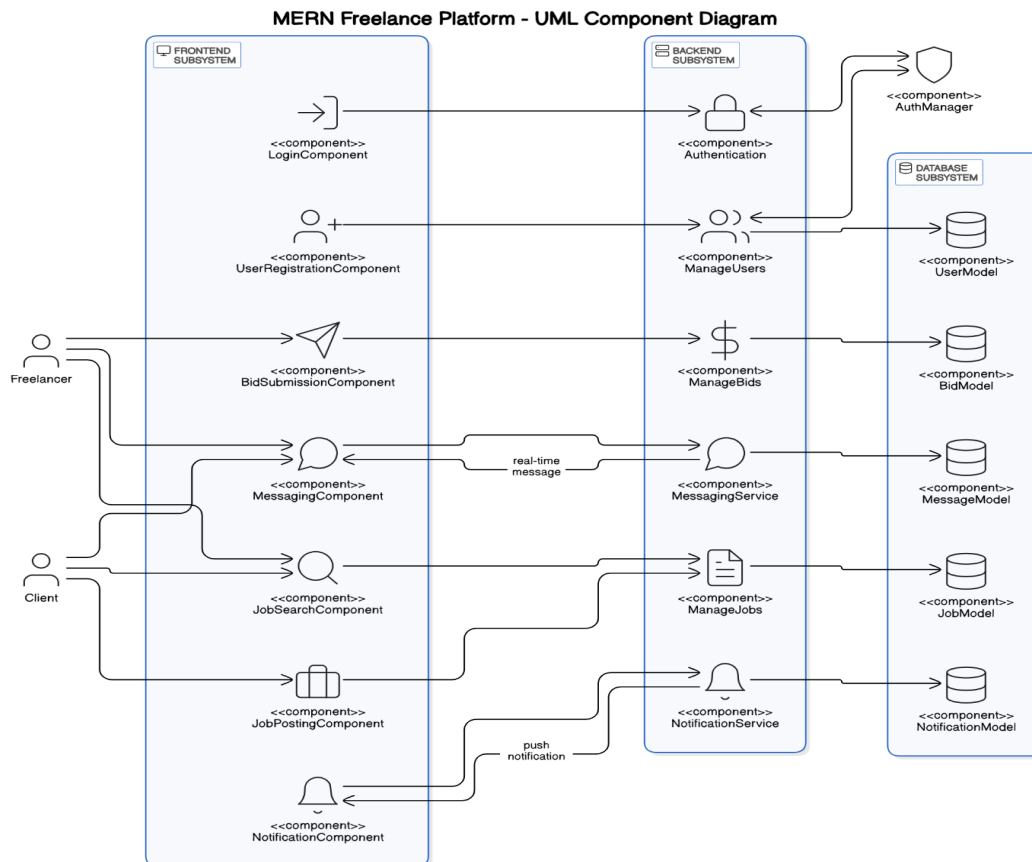
UML PACKAGE DIAGRAM



DEPLOYMENT DIAGRA



COMPONENT DIAGRAM:



C- TEST CASES BLACKBOX

3.1 Introduction

In this section, we present the Black Box Testing results for our Freelance Job Portal Application. Testing was conducted based on two major Black Box techniques:

- Equivalence Class Partitioning (ECP)
- Boundary Value Analysis (BVA)

Additionally, system functionality was validated based on the user stories defined during the development phase.

3.2 Test Case Generation

Equivalence Class Partitioning:

Test cases were generated by dividing input data into valid and invalid classes to ensure all scenarios are covered efficiently.

Boundary Value Analysis:

Critical boundaries of input fields (like minimum and maximum lengths for usernames, passwords, bid amounts, etc.) were tested.

3.3 Test Cases Summary

- Total number of Black Box test cases: 10+
- Techniques applied: Equivalence Class Partitioning, Boundary Value Analysis
- User stories covered: Profile Management, Job Posting, Job Search, Bidding, Rating, etc.

| | A | B | C | D | E | F |
|----|--------------|-------------------|--------------------------------------|----------------------------------------------|-----------------------------------------|-----------------------|
| 1 | Test Case ID | Use Case | Test Scenario | Input Data | Expected Result | Test Type |
| 2 | TC1 | User Registration | Register with valid details | Valid name, valid email (test@mail.com), pa | Registration successful | Equivalence Partition |
| 3 | TC2 | User Registration | Register with invalid email format | Email: testmail.com | Error: Invalid email format | Equivalence Partition |
| 4 | TC3 | User Registration | Register with password too short | Password: 123 | Error: Password must be at least 6 cha | Boundary Value |
| 5 | TC4 | User Registration | Register with invalid role | Role: admin | Error: Invalid role selected | Equivalence Partition |
| 6 | TC5 | Job Posting | Post job with valid data | Title: Web App, Budget: 500, client_id valid | Job posted successfully | Equivalence Partition |
| 7 | TC6 | Job Posting | Post job with zero budget | Budget: 0 | Error: Budget must be greater than 0 | Boundary Value |
| 8 | TC7 | Bid Submission | Submit bid with positive amount | Bid amount: 300 | Bid submitted successfully | Equivalence Partition |
| 9 | TC8 | Bid Submission | Submit bid with negative amount | Bid amount: -100 | Error: Bid amount must be greater tha | Equivalence Partition |
| 10 | TC9 | Submit Review | Submit valid rating | Rating: 5 | Review submitted successfully | Equivalence Partition |
| 11 | TC10 | Submit Review | Submit invalid rating | Rating: 6 | Error: Rating must be between 1 and 5 | Boundary Value |
| 12 | TC11 | Bookmark Job | Bookmark a job with valid freelancer | Correct freelancer_id and job_id | Bookmark added successfully | Equivalence Partition |
| 13 | TC12 | Bookmark Job | Bookmark with missing job_id | freelancer_id valid, job_id missing | Error: Freelancer ID and Job ID are req | Equivalence Partition |
| 14 | | | | | | |
| 15 | | | | | | |

3.4 Conclusion

- Black Box Testing confirmed that the system meets functional requirements as per the user stories.
- No critical errors were found during this phase.

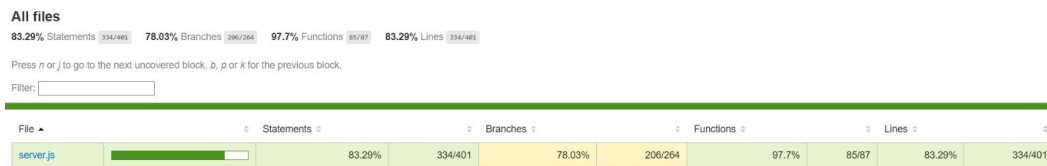
D- WHITEBOX TESTING

Tool Used

We used Jest, a JavaScript testing framework suitable for Node.js, to perform unit testing on key backend functions in server.js.

Coverage Metrics Achieved

| Coverage Type | Achieved | Total Lines/Units |
|---------------|----------|-------------------|
| Statements | 83.29% | 334/401 |
| Branches | 78.03% | 206/264 |
| Functions | 97.7% | 85/87 |
| Lines | 83.29% | 33 |



E- FINAL REPORT

1. Project Introduction

"New Balance" is a freelance job portal that connects clients and freelancers through a secure, feature-rich platform. Clients can post jobs, and freelancers can bid on them. Key features include registration/authentication, profile management, job bidding, real-time messaging, notifications, feedback system, and work tracking. The portal aims to simplify remote hiring and freelancing for both parties.

2. Functional and Non-Functional Requirements

Functional Requirements

- User registration and authentication
- Job posting and bidding system
- Profile management (client & freelancer)
- Messaging and notifications
- Rating and review system

- Admin panel for system management

Non-Functional Requirements

- High availability and scalability
- Secure password storage and session handling
- Responsive UI for mobile and desktop
- Performance optimization for job search and bid processing
- Robust error handling and validation

3. User Stories (Total 15)

Phase 1:

- As a client, I want to post a job so freelancers can apply.
- As a freelancer, I want to search for jobs based on skills.
- As a user, I want to log in securely.
- As a freelancer, I want to bid on a job.
- As a user, I want to edit my profile information.

Phase 2 :

- As a client, I want to receive notifications for new bids.
- As a freelancer, I want to bookmark interesting jobs.
- As a user, I want to message other users.
- As a freelancer, I want to see my work history.
- As a client, I want to rate freelancers after job completion.

Phase 3:

- As a user, I want to log out from my account.
- As a freelancer, I want to receive custom job offers.
- As an admin, I want to monitor and manage job posts.
- As a client, I want to manage ongoing job contracts.
- As a freelancer, I want to receive real-time notifications.

4. Product Backlog

| ID | User Story / Feature | Priority | Status |
|----|-----------------------------|----------|-----------|
| 1 | User Registration (Sign Up) | High | Completed |

| | | | |
|----|------------------------------------|--------|-------------------------|
| 2 | User Authentication (Login/Logout) | High | Completed |
| 3 | Profile Management | High | Completed |
| 4 | Job Posting | High | Completed |
| 5 | Job Search & Filtering | High | Completed |
| 6 | Bid Submission | High | Completed |
| 7 | Proposal Review & Approval | Medium | Completed |
| 8 | Freelancer Rating & Review | Medium | Completed |
| 9 | Client Rating & Review | Medium | Completed |
| 10 | Bookmarking Jobs | Medium | Completed |
| 11 | Job Completion & Feedback | High | Implemented in Sprint 3 |
| 12 | Custom Job Offer | Medium | Implemented in Sprint 3 |
| 13 | Work History & Earnings Tracking | Medium | Implemented in Sprint 3 |
| 14 | Real-Time Notification | Medium | Implemented in Sprint 3 |
| 15 | Messaging System | High | Implemented in Sprint 3 |

5. Sprin1 and sprint 2 backlog

Sprint 1 Backlog

| ID | User Story | Priority | Status |
|----|------------------------------------|----------|-----------|
| 1 | User Registration (Sign Up) | High | Completed |
| 2 | User Authentication (Login/Logout) | High | Completed |
| 3 | Profile Management | High | Completed |
| 4 | Job Posting | High | Completed |
| 5 | Job Search & Filtering | High | Completed |

Sprint 2 Backlog

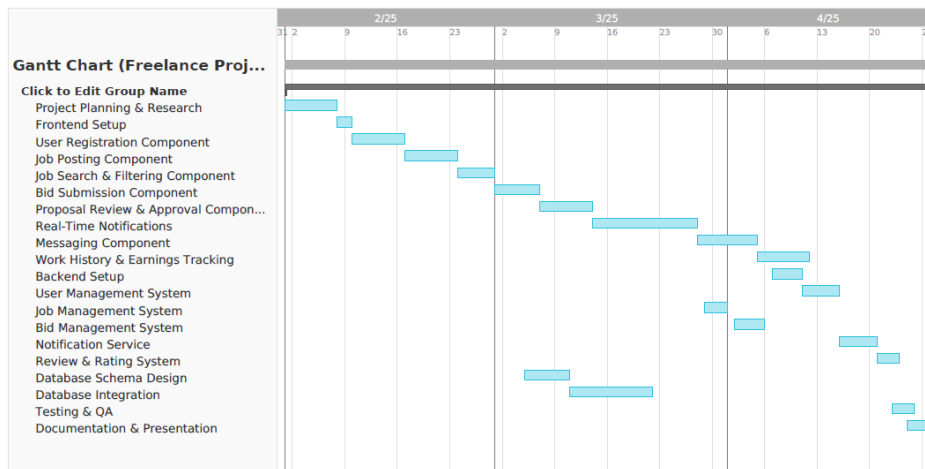
| ID | User Story | Priority | Status |
|----|----------------------------|----------|-----------|
| 6 | Bid Submission | High | Completed |
| 7 | Proposal Review & Approval | Medium | Completed |
| 8 | Freelancer Rating & Review | Medium | Completed |
| 9 | Client Rating & Review | Medium | Completed |
| 10 | Job Completion & Feedback | High | Completed |

6. Project Plan

The project is structured into five major phases: Initiation, Frontend Development, Backend Development, Testing, and Maintenance. It begins with requirements gathering, use case analysis, technology stack selection, and initial wireframe design. Frontend development

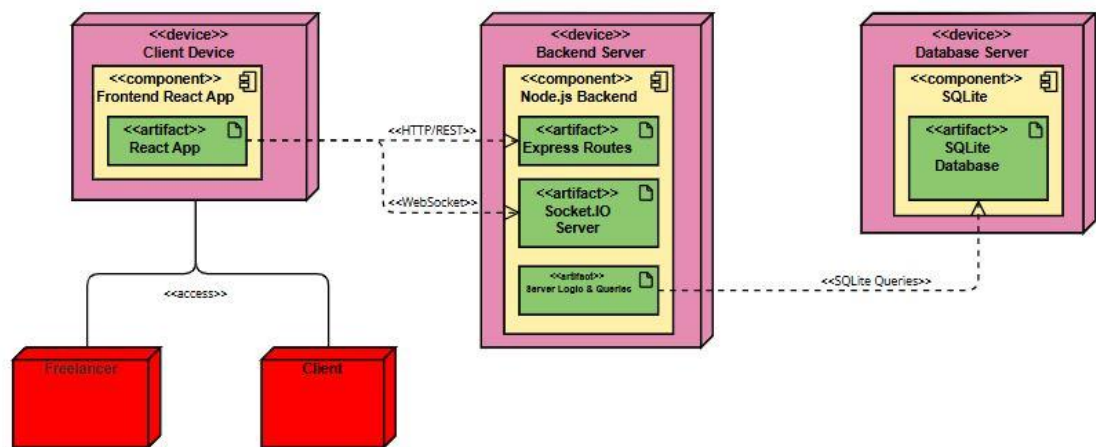
includes building user-friendly interfaces for registration, dashboards, job postings, bidding, profiles, messaging, and notifications. Backend development involves designing a MongoDB schema and creating secure REST and WebSocket APIs for user management, job handling, bidding, messaging, feedback, and admin control. Testing includes unit, integration, and user acceptance testing to ensure system quality and usability. The final phase focuses on ongoing maintenance, bug fixing, feature enhancements, and system monitoring.

teamgantt
Created with Free Edition



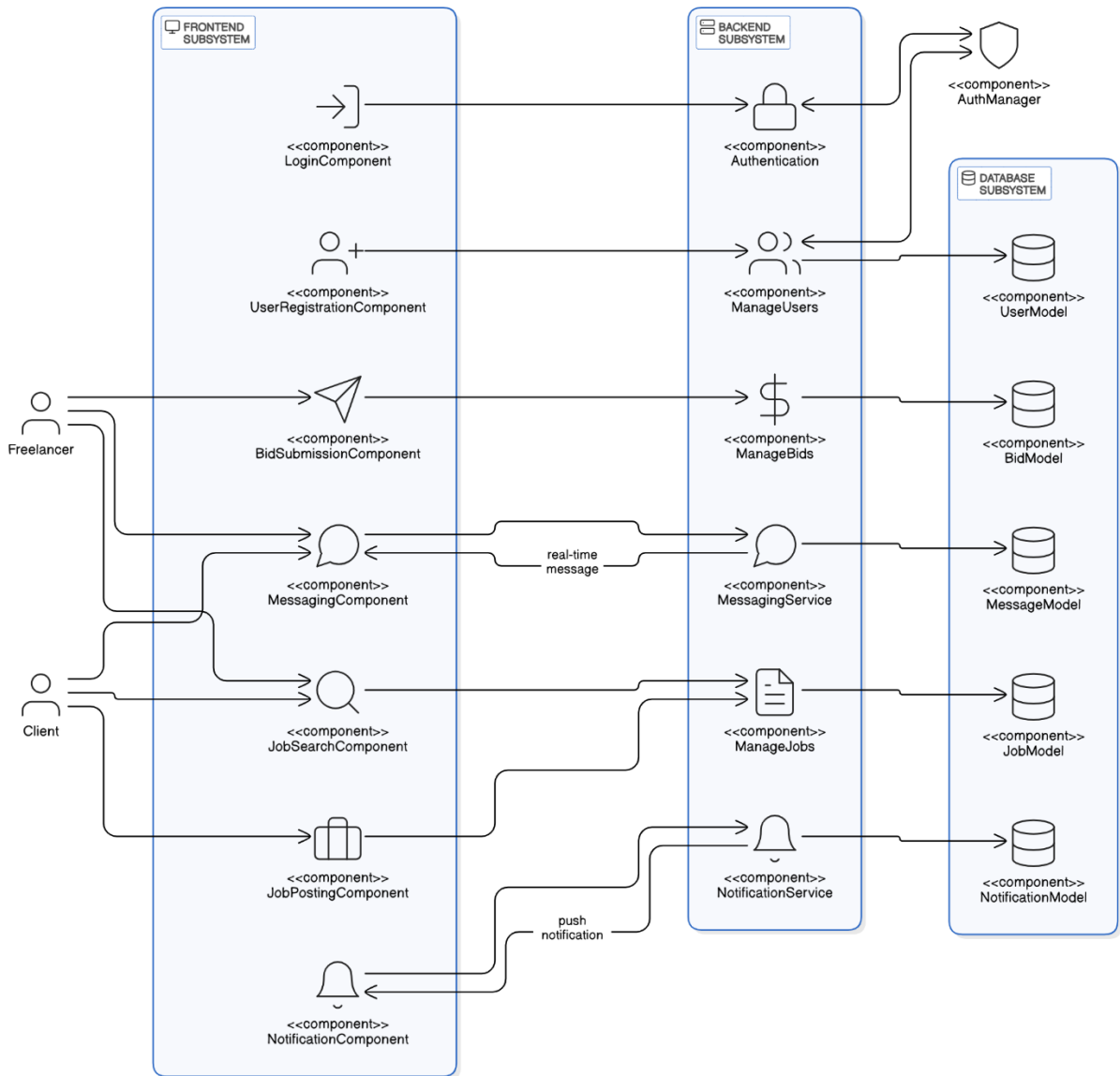
7. Architecture Diagrams

DEPLOYMENT DIAGRAM:



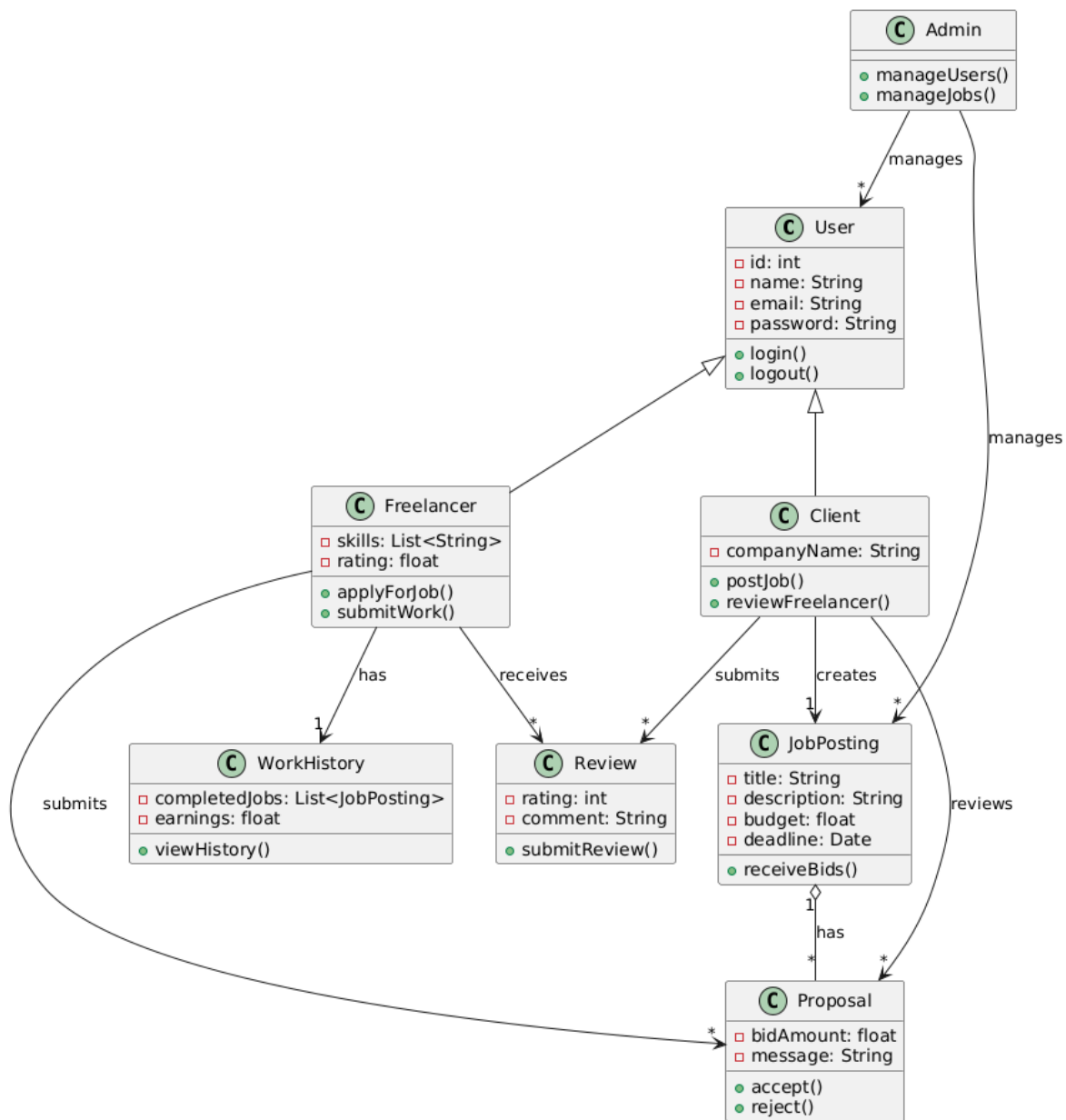
COMPONENT DIAGRAM:

MERN Freelance Platform - UML Component Diagram

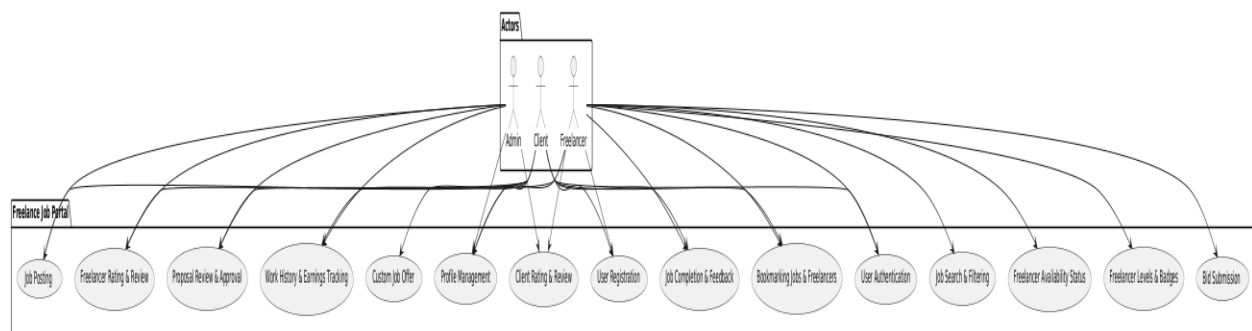


8. Design

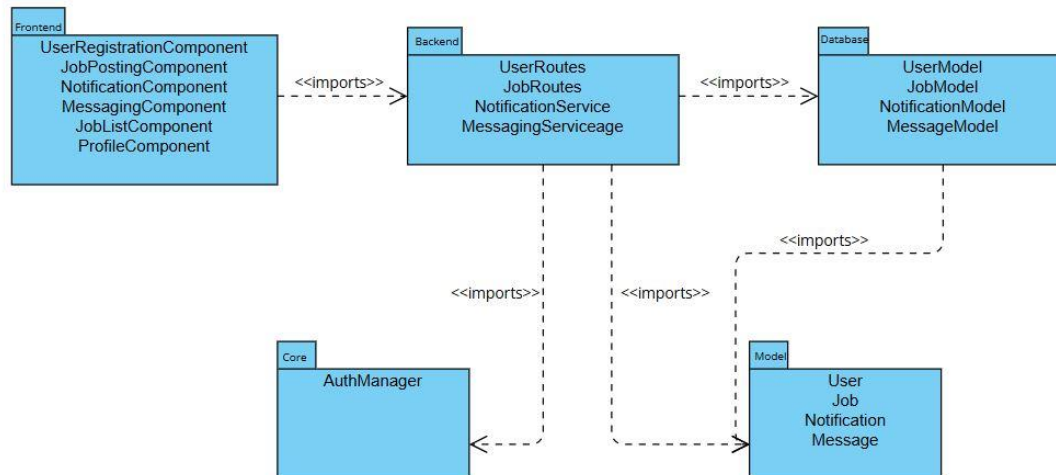
Class Diagram:



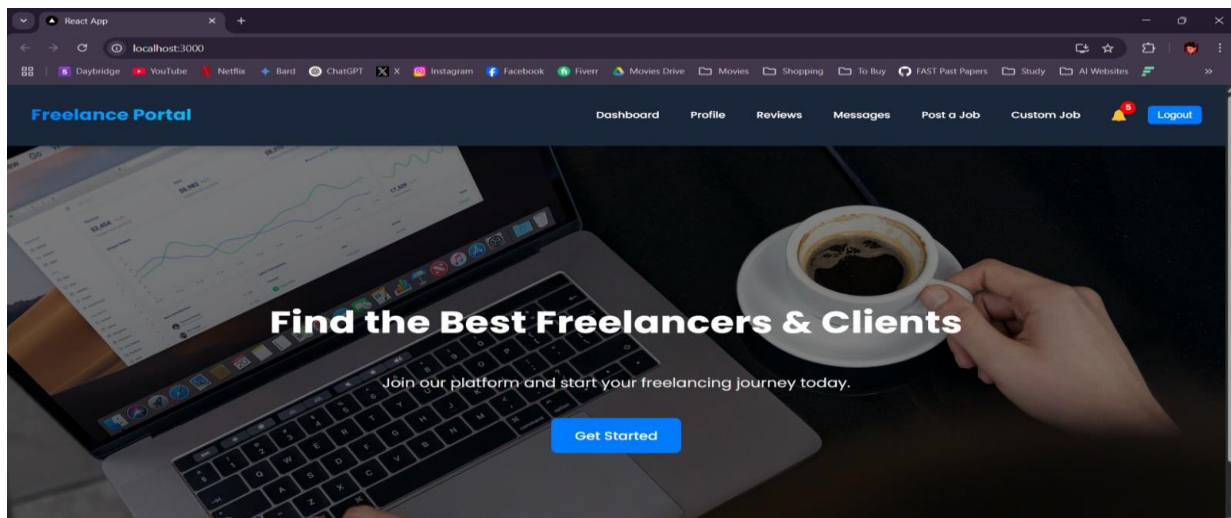
Usecase Diagram:



UML Package Diagram:




9. Actual implementation Screenshots



React App

localhost:3000/signup

Daybridge YouTube Netflix Bard ChatGPT X Instagram Facebook Fiverr Movies Drive Movies Shopping To Buy FAST Past Papers Study AI Websites

Freelance Portal Dashboard Profile Reviews Messages Post a Job Custom Job  Logout

Signup

Full Name

Email

Password

Freelancer


Signup

Already have an account? [Login](#)

React App

localhost:3000/job-posting

Daybridge YouTube Netflix Bard ChatGPT X Instagram Facebook Fiverr Movies Drive Movies Shopping To Buy FAST Past Papers Study AI Websites

Freelance Portal Dashboard Profile Reviews Messages Post a Job Custom Job  Logout

Post a Job

Job Title

New Job

Job Description

New job description

Budget (\$)

1000

Post Job

React App

localhost:3000/client-dashboard

Daybridge YouTube Netflix Bard ChatGPT X Instagram Facebook Fiverr Movies Drive Movies Shopping To Buy FAST Past Papers Study AI Websites

Freelance Portal Dashboard Profile Reviews Messages Post a Job Custom Job 5 Logout

Client Dashboard

Your Posted Jobs

New Job

New job description

Budget: \$1000.00

[View Proposals](#)

Ongoing Jobs

No ongoing jobs.

React App

localhost:3000/reviews

Daybridge YouTube Netflix Bard ChatGPT X Instagram Facebook Fiverr Movies Drive Movies Shopping To Buy FAST Past Papers Study AI Websites

Freelance Portal Dashboard Profile Reviews Messages Post a Job Custom Job 5 Logout

Jobs to Review

New job for fl

Freelancer: Asjad Free 1

Amount: \$400

Rating (1-5)

Write a review...

[Submit Review](#)

non custom job

Freelancer: Asjad Free 1

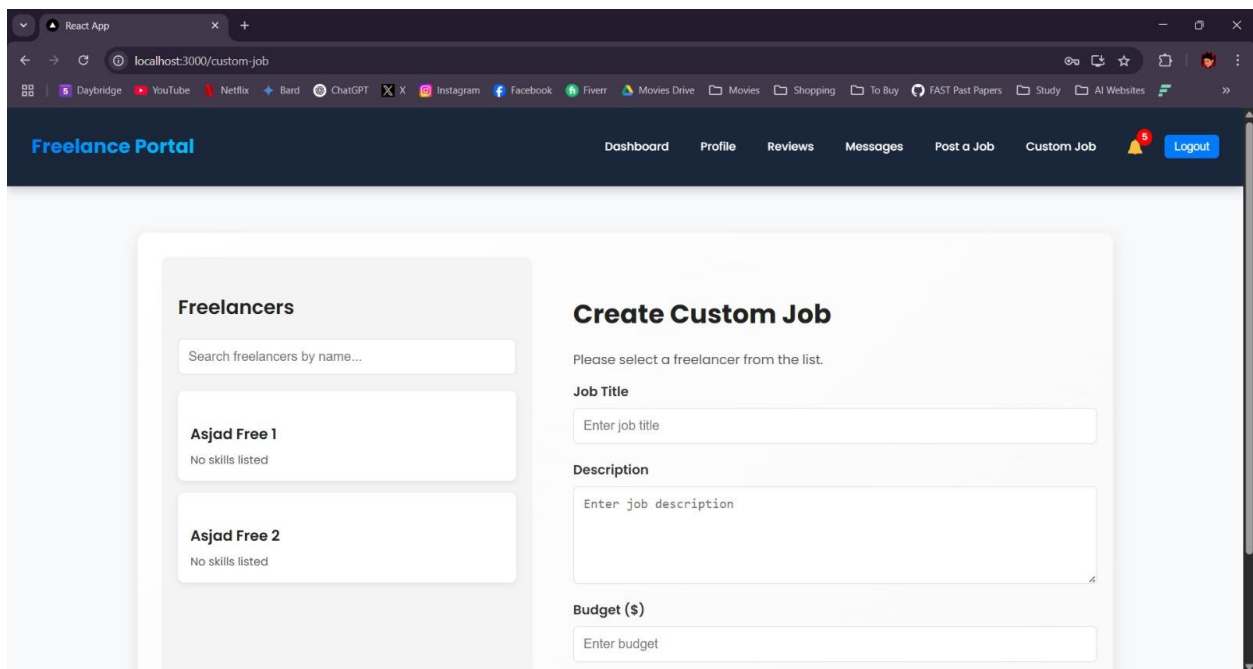
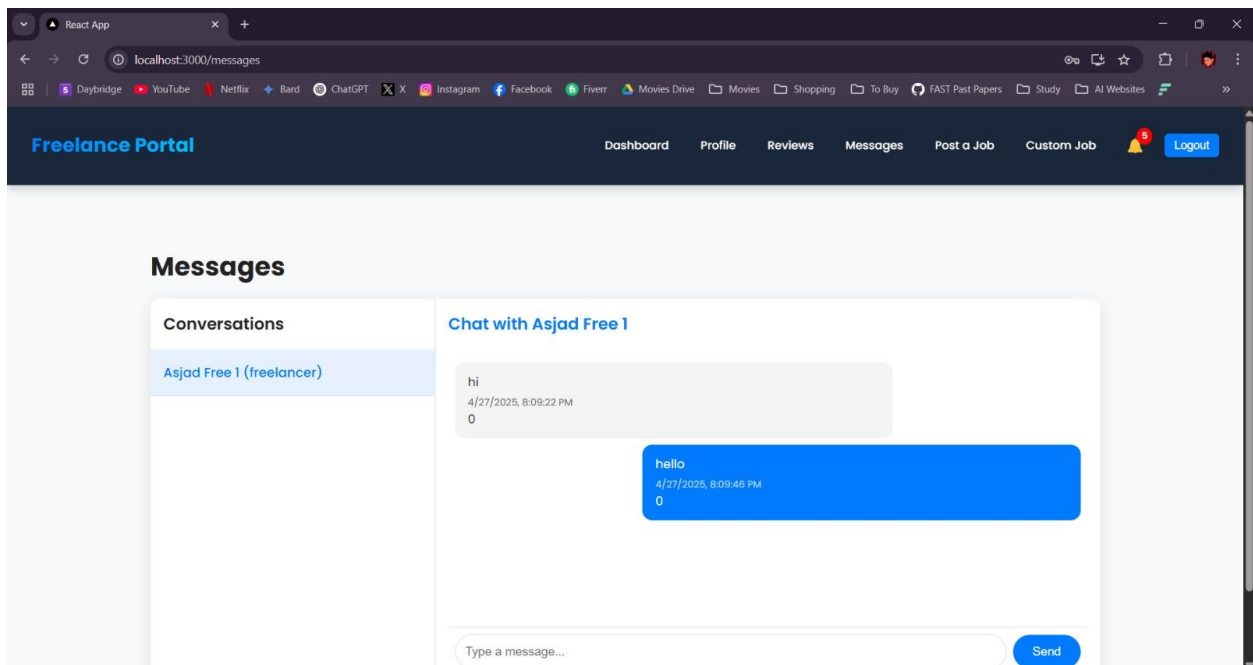
Amount: \$500

Rating (1-5)

Write a review...

[Submit Review](#)

Reviews Received

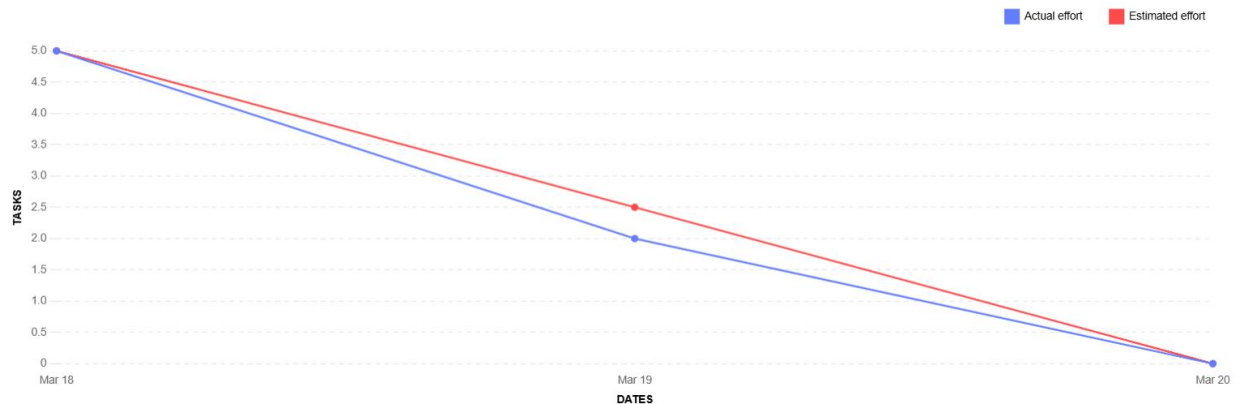


10. Product Burn down chart for sprint 1 and 2

Sprint1 :

Burndown chart
Sprint 1

[Edit sprint](#)

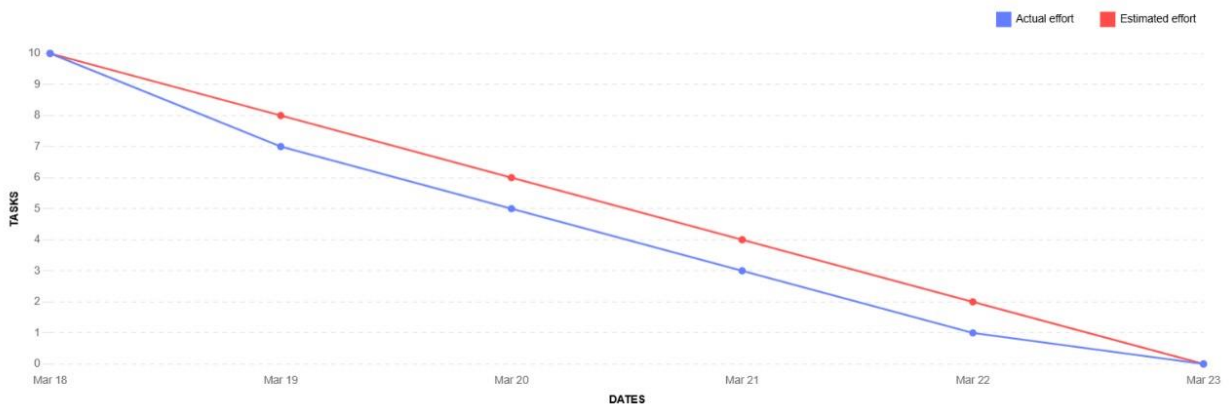


Sprint2:

Burndown chart

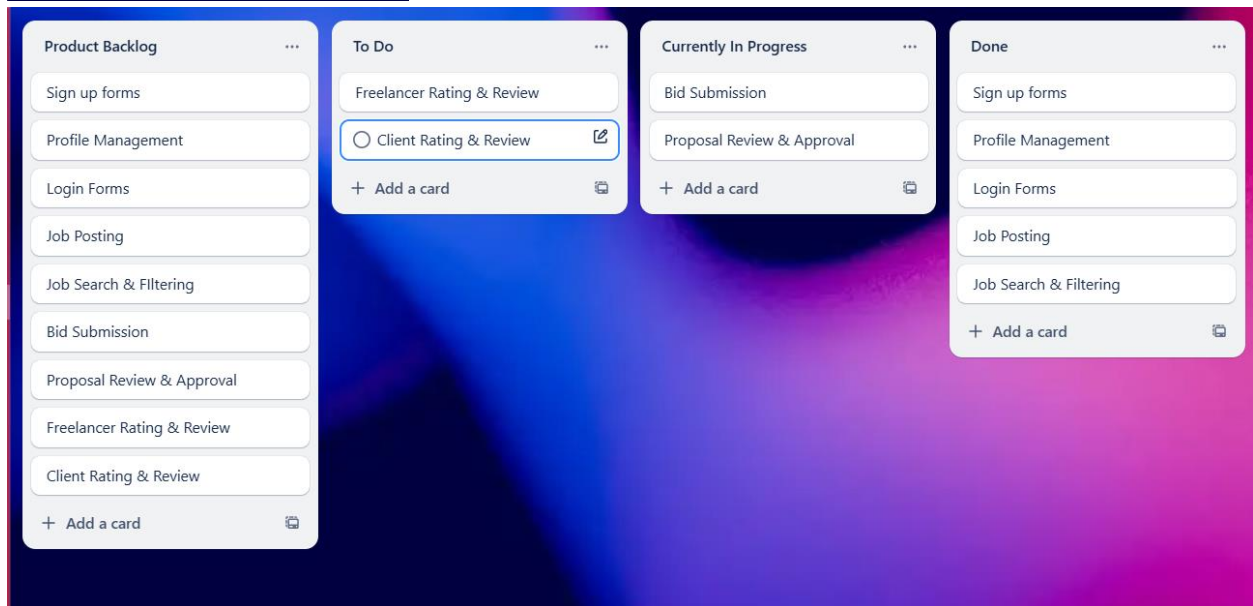
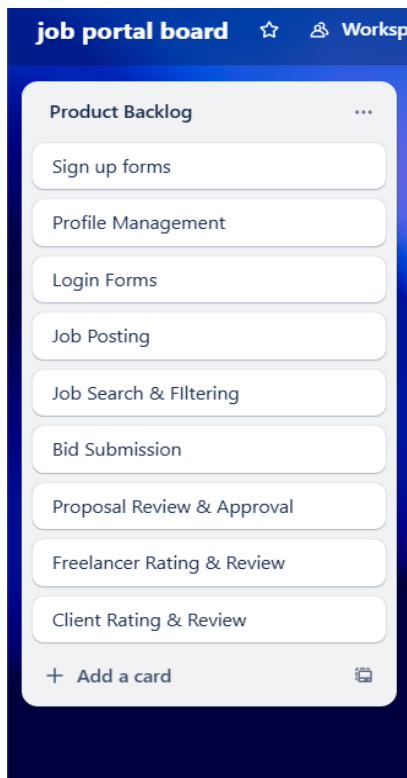
Product Backlog Burndown Chart

[Edit sprint](#)



11. Trello board screen shots

Product Backlog:



Sprint Backlog:

Sprint 1:

The image displays two screenshots of a Jira Kanban board titled "job portal board". The board is organized into five columns: "Sprint 1 Backlog", "To Do", "In Progress", "Completed", and "Sprint 2 Backlog".

Top Screenshot:

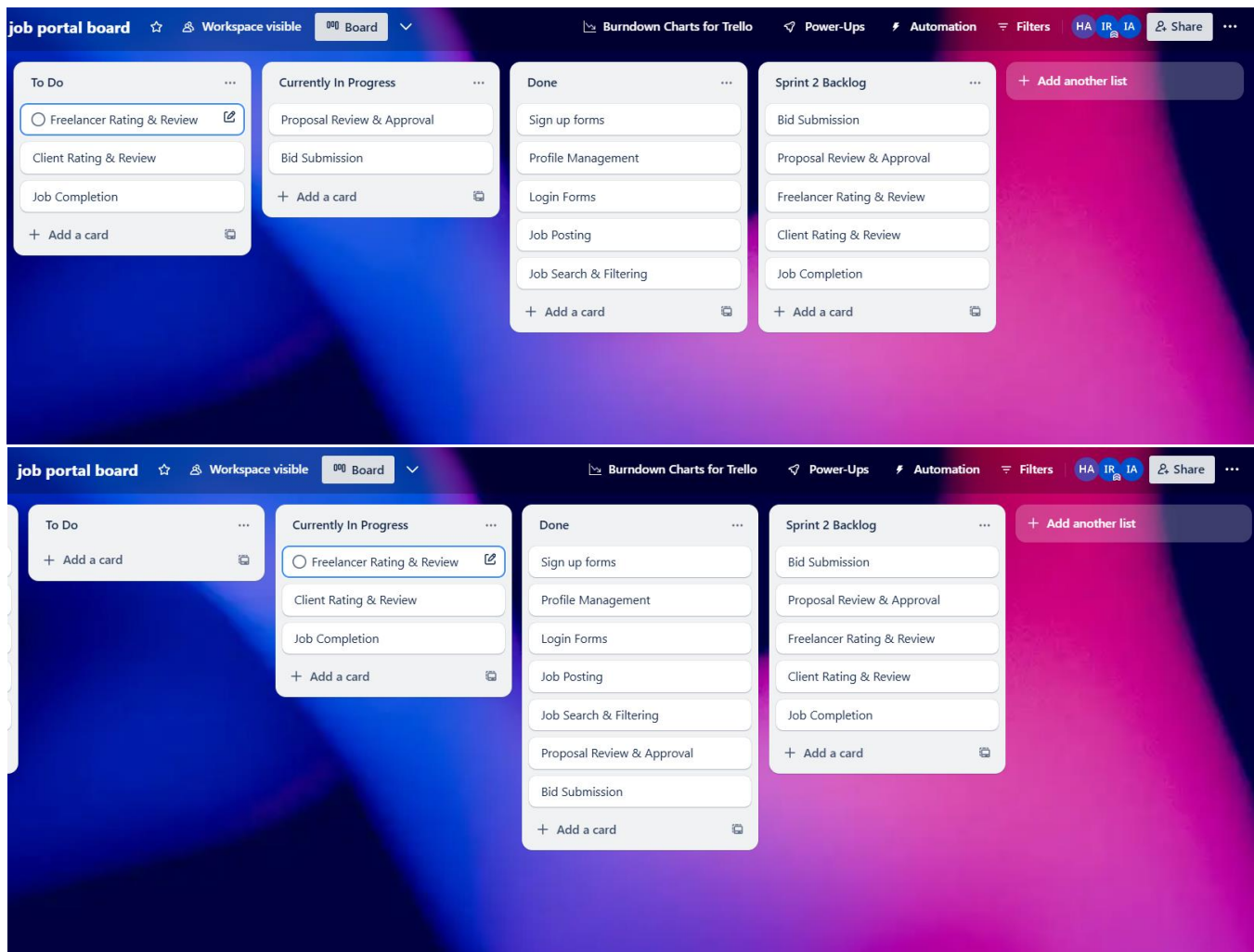
- Sprint 1 Backlog:** Contains five cards: "Sign Up forms", "Login Forms", "Profile Management", "Job Posting", and "Job Search & Filtering".
- To Do:** Contains four cards: "Sign up forms", "Login Forms", "Profile Management", and "Job Posting".
- In Progress:** Empty, with a "+ Add a card" button.
- Completed:** Empty, with a "+ Add a card" button.
- Sprint 2 Backlog:** Contains five cards: "Bid Submission", "Proposal Review & Approval", "Freelancer Rating & Review", "Client Rating & Review", and "Bookmarking Jobs & Freelancers".

Bottom Screenshot:

- Sprint 1 Backlog:** Contains five cards: "Sign Up forms", "Login Forms", "Profile Management", "Job Posting", and "Job Search & Filtering".
- To Do:** Contains one card: "Job Search & Filtering".
- In Progress:** Contains two cards: "Profile Management" and "Job Posting".
- Completed:** Contains two cards: "Sign up forms" and "Login Forms".
- Sprint 2 Backlog:** Contains five cards: "Bid Submission", "Proposal Review & Approval", "Freelancer Rating & Review", "Client Rating & Review", and "Bookmarking Jobs & Freelancers".

The interface includes a top navigation bar with "Power-Ups", "Automation", "Filters", and "Share" options. The board is set to "Board" view and "Workspace visible".

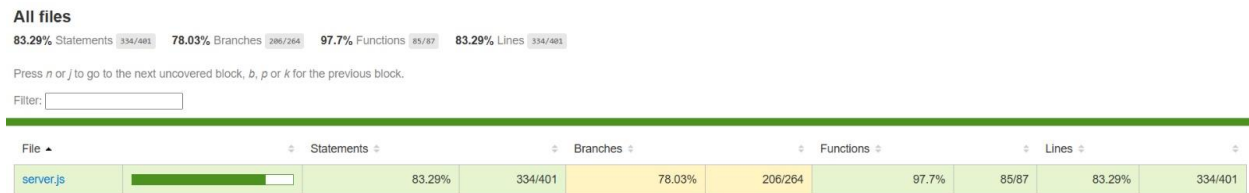
Sprint 2:



12. Testcases -Black box

| | A | B | C | D | E | F |
|----|--------------|-------------------|--------------------------------------|----------------------------------------------|-----------------------------------------|-----------------------|
| 1 | Test Case ID | Use Case | Test Scenario | Input Data | Expected Result | Test Type |
| 2 | TC1 | User Registration | Register with valid details | Valid name, valid email (test@mail.com), pa | Registration successful | Equivalence Partition |
| 3 | TC2 | User Registration | Register with invalid email format | Email: testmail.com | Error: Invalid email format | Equivalence Partition |
| 4 | TC3 | User Registration | Register with password too short | Password: 123 | Error: Password must be at least 6 cha | Boundary Value |
| 5 | TC4 | User Registration | Register with invalid role | Role: admin | Error: Invalid role selected | Equivalence Partition |
| 6 | TC5 | Job Posting | Post job with valid data | Title: Web App, Budget: 500, client_id valid | Job posted successfully | Equivalence Partition |
| 7 | TC6 | Job Posting | Post job with zero budget | Budget: 0 | Error: Budget must be greater than 0 | Boundary Value |
| 8 | TC7 | Bid Submission | Submit bid with positive amount | Bid amount: 300 | Bid submitted successfully | Equivalence Partition |
| 9 | TC8 | Bid Submission | Submit bid with negative amount | Bid amount: -100 | Error: Bid amount must be greater tha | Equivalence Partition |
| 10 | TC9 | Submit Review | Submit valid rating | Rating: 5 | Review submitted successfully | Equivalence Partition |
| 11 | TC10 | Submit Review | Submit invalid rating | Rating: 6 | Error: Rating must be between 1 and 5 | Boundary Value |
| 12 | TC11 | Bookmark Job | Bookmark a job with valid freelancer | Correct freelancer_id and job_id | Bookmark added successfully | Equivalence Partition |
| 13 | TC12 | Bookmark Job | Bookmark with missing job_id | freelancer_id valid, job_id missing | Error: Freelancer ID and Job ID are req | Equivalence Partition |
| 14 | | | | | | |
| 15 | | | | | | |

13. Testcases -White box



14. Work Division between group members

Task Member Responsible

| | |
|---------------------------------|---------------|
| Frontend: UI Design & Dashboard | Raza Khan |
| Backend: API Development | Hassaan Afzal |
| Database & Schema Design | AsjadUllah |
| Testing: Black Box | AsjadUllah |
| Testing: White Box | Hassaan Afzal |
| Documentation & Report Writing | Raza Khan |

15. Lessons Learned by Group

- Learned how to manage a real-world project using Agile practices.
- Understood the value of user stories and sprints in iterative development.
- Discovered the importance of test coverage in maintaining code quality.
- Learned to divide tasks effectively using tools like Trello and Gantt charts.
- Gained experience using full-stack technologies (React, Node.js, MongoDB).