



# ***North South University***

***Department of Electrical & Computer Engineering***

**Project Report  
for  
Intelligent Task Planner for Students  
(BrainyBalance)**

**Submitted to:  
Dr. Mohammad Rezwanul Huq(MRH1)  
Associate Professor**

**Group No. - 06**

**Group Members:**

<b>ID</b>	<b>Name</b>
<b>2312561642</b>	<b>Pujon Chakraborty</b>
<b>2212650042</b>	<b>Ifrat Afroze Niniya</b>
<b>2222756042</b>	<b>Sudipto Roy</b>
<b>2311539042</b>	<b>Arafat Zaman Ratul</b>

**Short Video Demonstration Link -**

[https://drive.google.com/file/d/1ti6AlJlx4DdWuZHUge2xVqBgl\\_EFsEWJ/view?usp=sharing](https://drive.google.com/file/d/1ti6AlJlx4DdWuZHUge2xVqBgl_EFsEWJ/view?usp=sharing)

# Software Requirements Specification

## 1. Introduction

### 1.1 Purpose

This SRS describes the functional and non-functional requirements for the Intelligent Task Planner for Students, a web-based productivity tool to assist university students in managing their tasks, assignments, exams, and study schedules.

### 1.2 Document Conventions

- Bold section headers.
- Requirements are labeled as REQ-x.

### 1.3 Project Scope

The project aims to create a scheduling assistant that utilizes rule-based or AI-driven logic to recommend optimal plans based on priority, deadlines, and user availability. It will provide visual analytics, reminders, and an intuitive UI.

### 1.4 Intended Audience and Reading Suggestions

This document is intended for developers, project supervisors, testers, and stakeholders. Readers should first review the Introduction and Overall Description before moving on to detailed features and requirements.

### 1.5 References

- TickTick: [https://ticktick.com/?language=en\\_US](https://ticktick.com/?language=en_US)
- IEEE SRS Standard

## 2. Overall Description

### 2.1 Product Perspective

The Intelligent Task Planner for Students is an independent web-based system. It shares conceptual similarities with productivity tools like and adds academic-specific features and intelligent time allocation logic.

### 2.2 Product Features

- Task creation and prioritization
- Smart scheduling recommendations
- Visual dashboard (calendar, charts)
- Notifications and reminders
- Pomodoro timer

## 2.3 User Classes and Characteristics

- Students: Primary users
- Guest User

## 2.4 Operating Environment

- Web browser (Chrome, Firefox, Brave etc.)
- Backend: Node.js
- Database: MongoDB

## 2.5 Design and Implementation Constraints

- Must use relational DBMS
- Optional AI scheduler must be modular
- Web technologies only

## 2.6 Assumptions and Dependencies

It is assumed that the Intelligent Task Planner for Students will be developed as a web-based application using standard technologies such as HTML, Tailwind CSS, and JavaScript for the frontend, with Node.js as the backend framework. The backend will be connected to a relational database system such as MongoDB to store user data, tasks, schedules, and preferences. The application requires users to have access to a stable internet connection to ensure smooth interaction with the system, particularly for features such as schedule generation and notification delivery. It is also assumed that users will provide accurate inputs regarding their tasks and available time blocks, as the scheduling logic depends on this data. The initial version of the scheduler will follow rule-based logic; however, there may be optional enhancements using AI techniques, which will depend on the availability of relevant libraries and training data. Integration with external APIs, such as Google Calendar, is optional and subject to proper API access and authentication credentials. Additionally, since the web application is designed primarily for student use, it is expected to handle moderate-scale user loads without the need for enterprise-grade infrastructure.

## 2.7 Stakeholders and Expected Benefits

- Students (Primary Users):
  - Role: Input tasks, deadlines, priorities, and available time; receive and adjust optimized schedules; access reminders and optional analytics.
  - Benefits: Improved task organization, reduced stress, and enhanced productivity through personalized schedules, timely notifications, and insights into time management efficiency.
- System Scheduler (AI/Rule-Based Engine):
  - Role: Processes user inputs to generate optimized task schedules and adapts to manual changes.
  - Benefits: Automates scheduling to deliver efficient, tailored plans, minimizing user effort and improving task completion rates.

# 3. System Features and Requirements

## 3.1 Functional Requirements

### 3.1 User Authentication

**Description:** Register new users and allow secure login/logout.

**Priority:** High

- REQ-1: The user shall be able to register and log in securely using a unique email and password.

### **3.2 Task Management**

**Description:** Add/edit/delete tasks with deadlines, priority, and tags.

**Priority:** High

- REQ-2: The user shall be able to create, edit, and delete tasks with attributes such as title, description, deadline, and priority.

### **3.3 Time Block Management**

**Description:** Define user availability to assist the scheduling engine.

**Priority:** High

- REQ-3: The user shall be able to specify available time blocks for scheduling tasks.

### **3.4 Scheduler Engine**

**Description:** Suggests an optimized schedule based on task data and time availability.

**Priority:** High

- REQ-4: The system shall generate a recommended task schedule using rule-based or AI-assisted logic.

### **3.5 Schedule Adjustment**

**Description:** Enable manual modification of generated schedules.

**Priority:** Medium

- REQ-5: The user shall be able to accept, adjust, or override the system-generated schedule.

### **3.6 Reminders and Notifications**

**Description:** Notify users about upcoming tasks.

**Priority:** Medium

- REQ-6: The system shall send reminders 15 minutes before scheduled tasks.

### **3.7 Visualization Dashboard**

**Description:** Display task distribution and time usage analytics.

**Priority:** Low

- REQ-7: The user shall be able to view daily and weekly visualizations such as calendars and charts.

## **3.2 System Features**

## **Authentication**

- Secure login/logout
- Session management for active users

## **Task Manager**

- CRUD operations on tasks
- Task fields: Title, Description, Deadline, Priority, Category

## **Scheduler Engine**

- Generates a daily/weekly schedule
- Respects available time, deadlines, and task priorities
- Allows manual edits by the user

## **Reminder System**

- Configurable task notifications
- Browser/email notifications

## **Pomodoro Timer**

- Timed intervals for task execution
- Visual indicator and logging for completed Pomodoros

## **Analytics Dashboard**

- Pie chart of time spent on tasks
- Weekly trend line for focus/productivity

# 3.3 External Interface Requirements

## **User Interfaces**

- Responsive web interface
- Calendar view, task input form, analytics dashboard

## **Hardware Interfaces**

- Not applicable (client-side browser)

## **Software Interfaces**

- Google Calendar API (optional)
- Database: MySQL

## **Communications Interfaces**

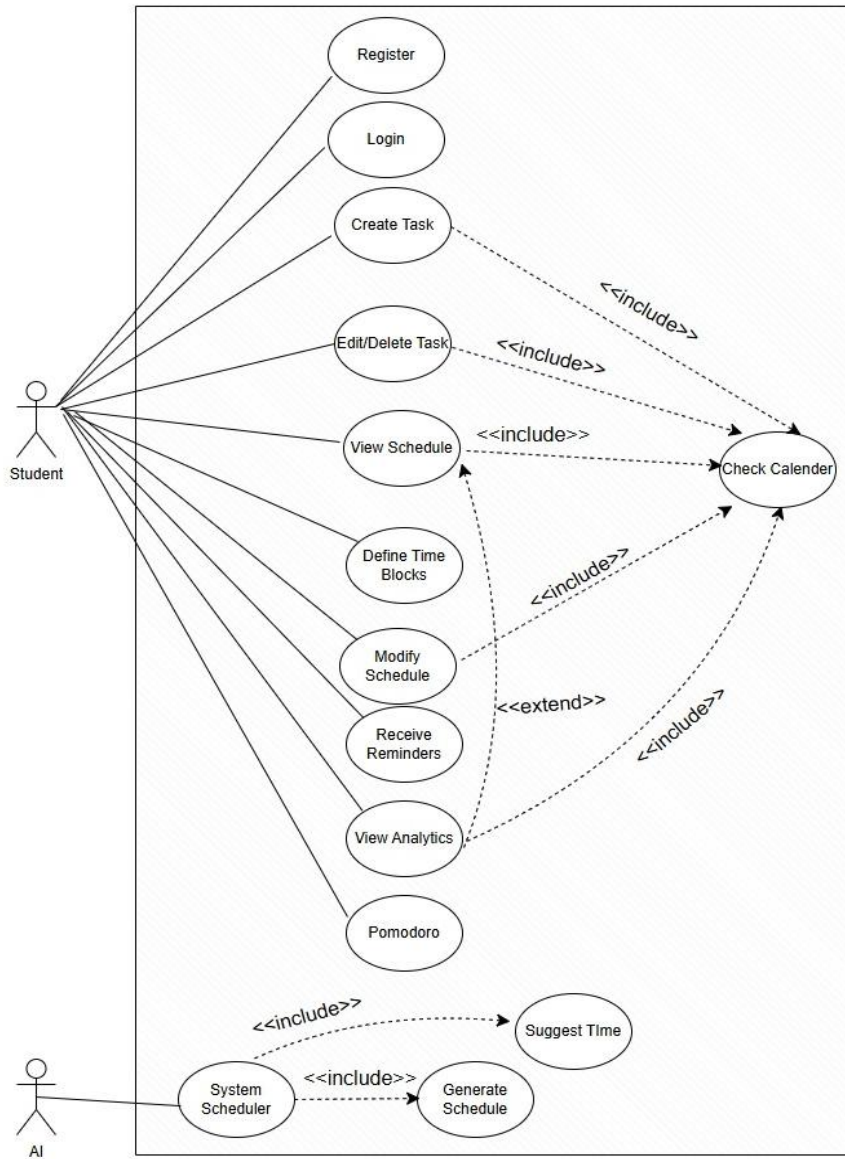
- HTTP/HTTPS for client-server communication

# 3.3 Non-Functional Requirements

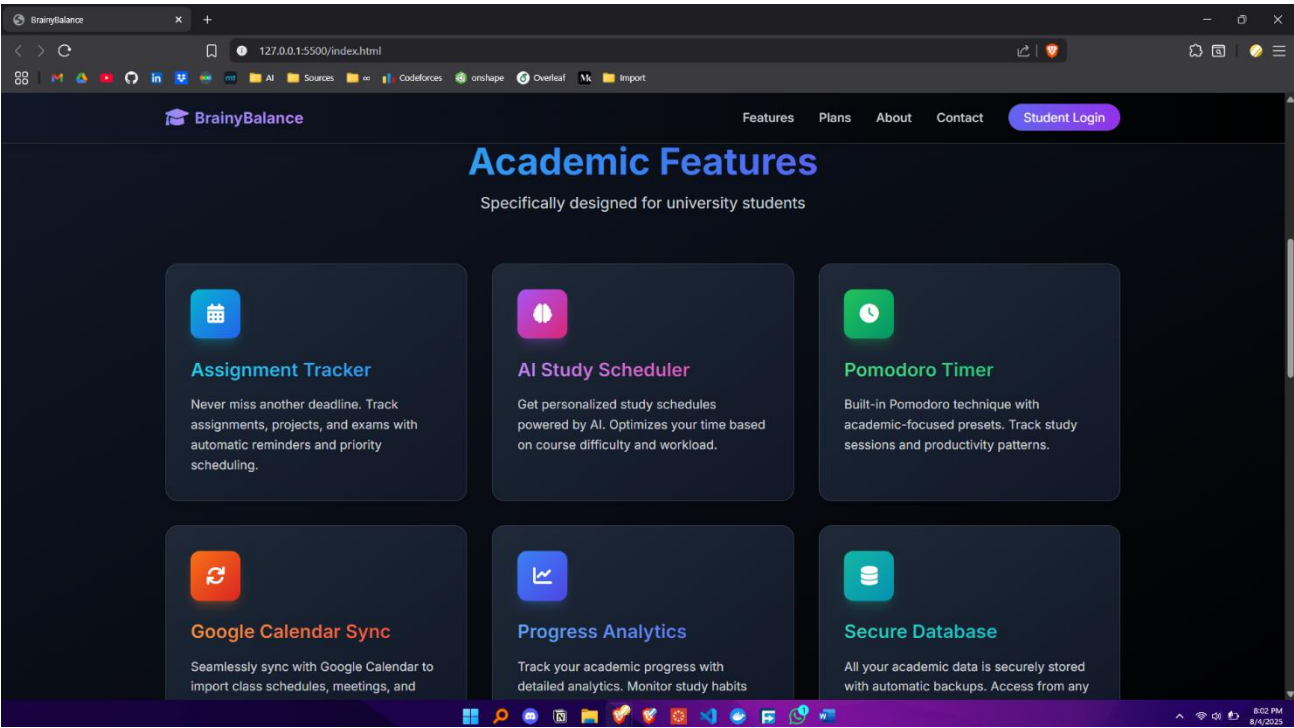
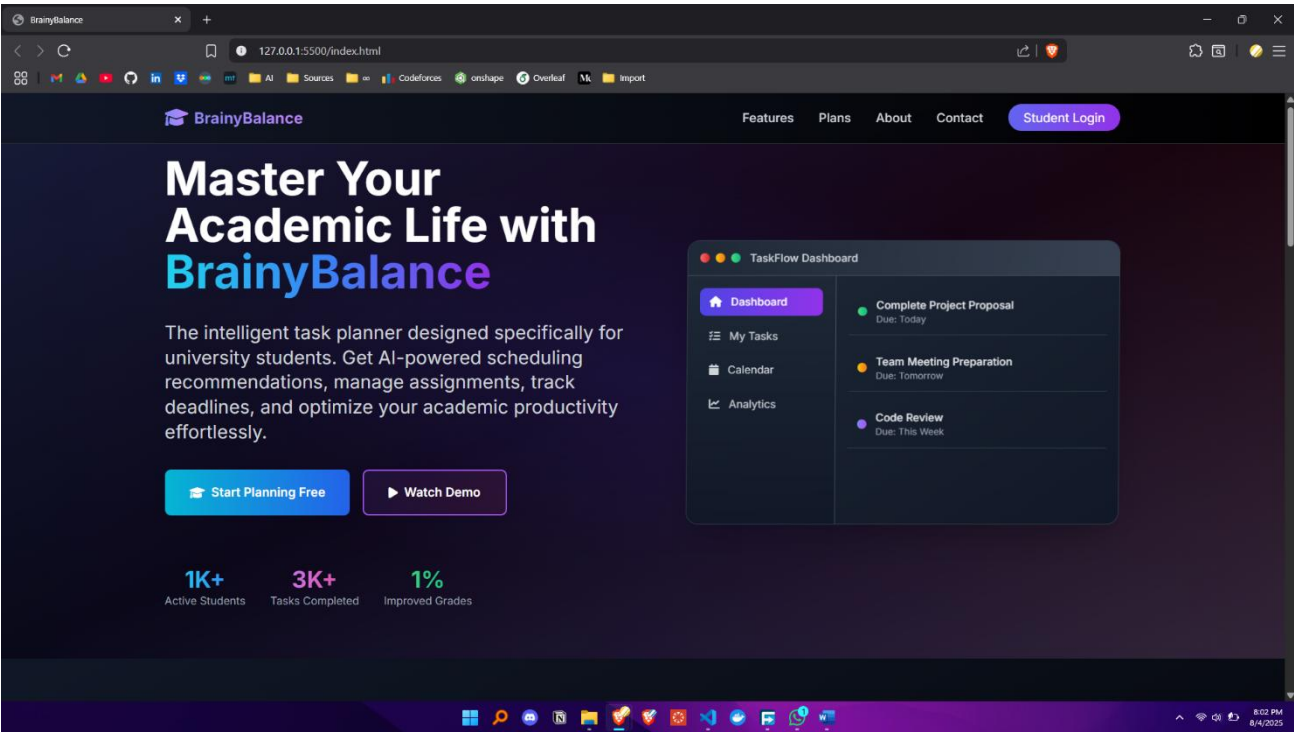
- **Security:** Password hashing, access control
- **Performance:** Task schedule generated under 2 seconds
- **Usability:** Simple, student-friendly UI
- **Scalability:** Must support multiple users concurrently
- **Reliability:** Auto-save and error recovery mechanisms

- **Maintainability:** Modular design for easy update
- **Compatibility:** Support multiple devices and browsers

## 4. System Use-Case Diagram



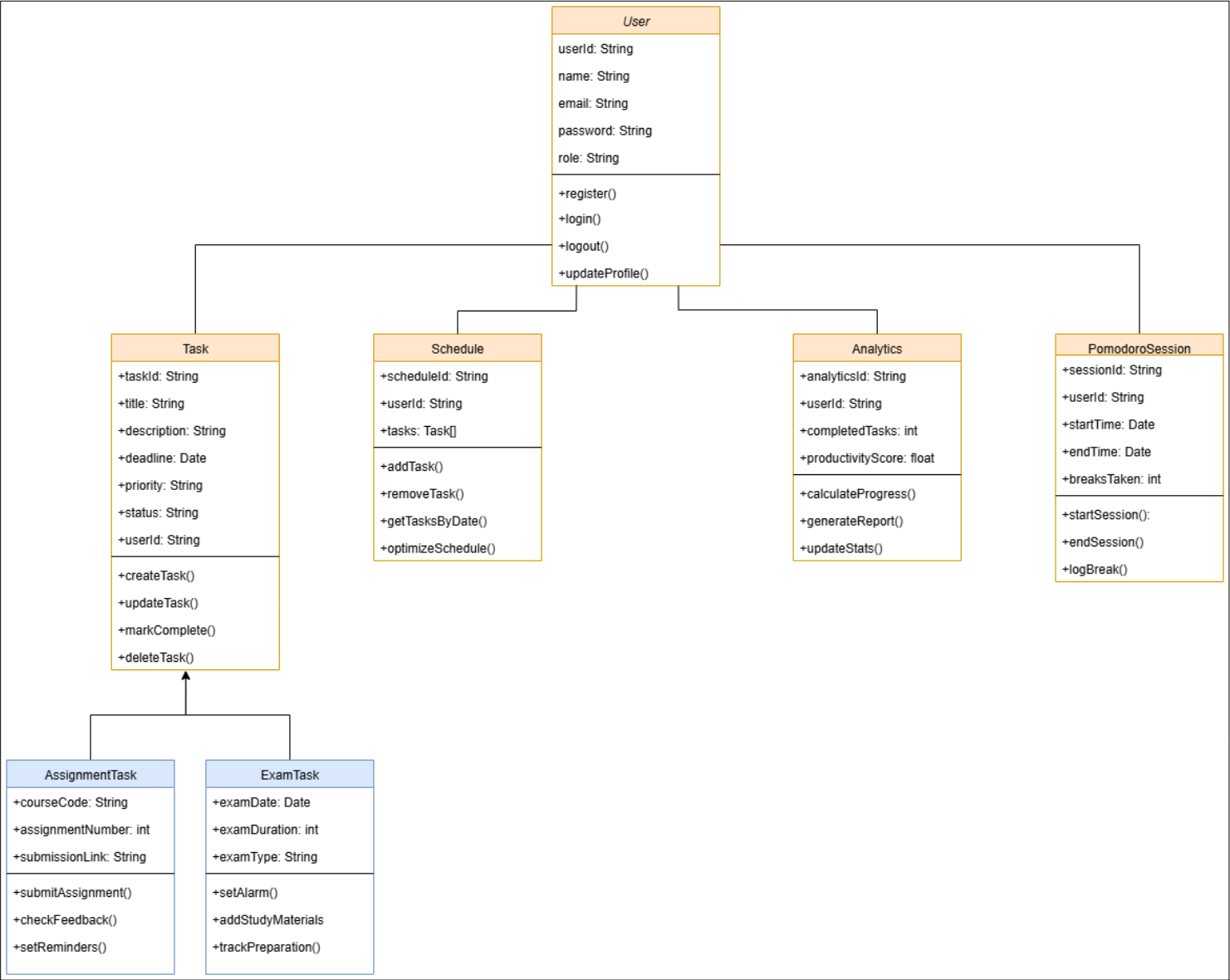
# 5. System Interface Mockups



# System Modeling Diagrams

## 1. Class Diagram:

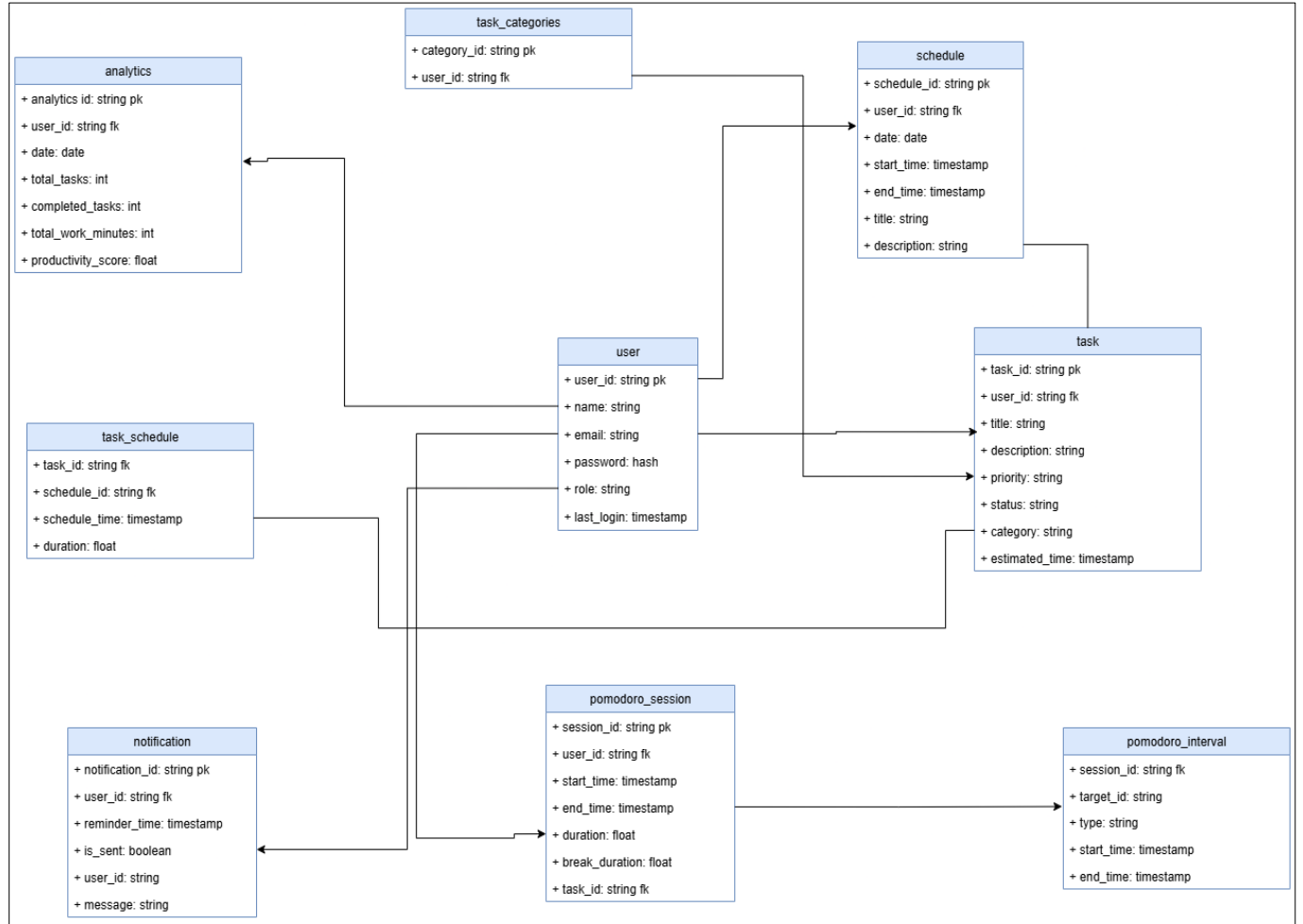
Contributor: Ifrat Afroze Niniya (ID: 2212650042)





## 2. ER Diagram:

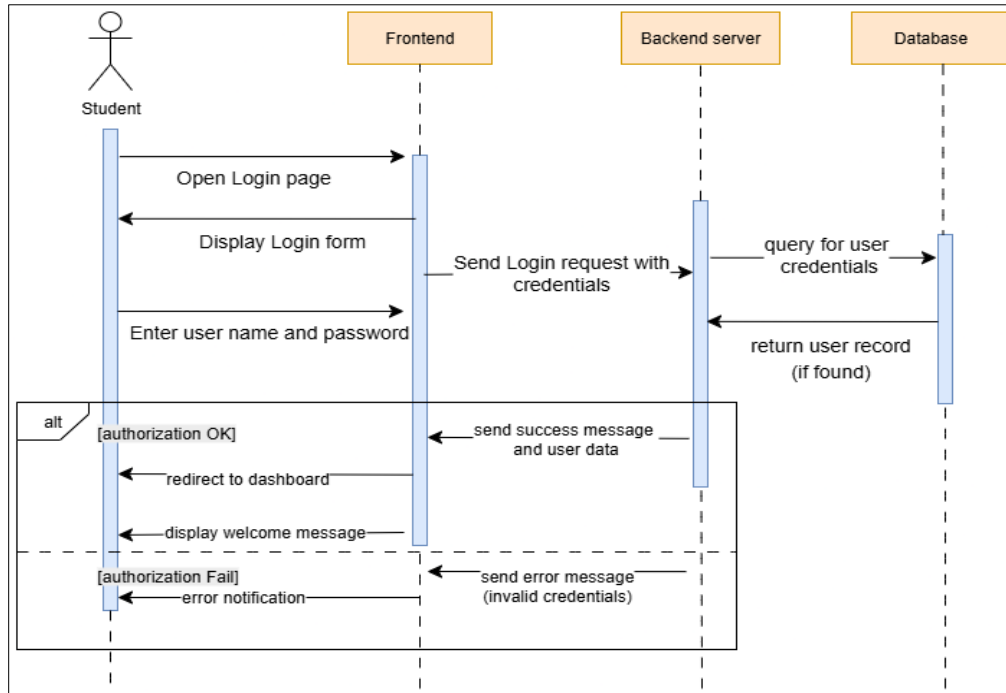
Contributor: Arafat Zaman Ratul (ID: 2311539042)



### 3. Sequence Diagram

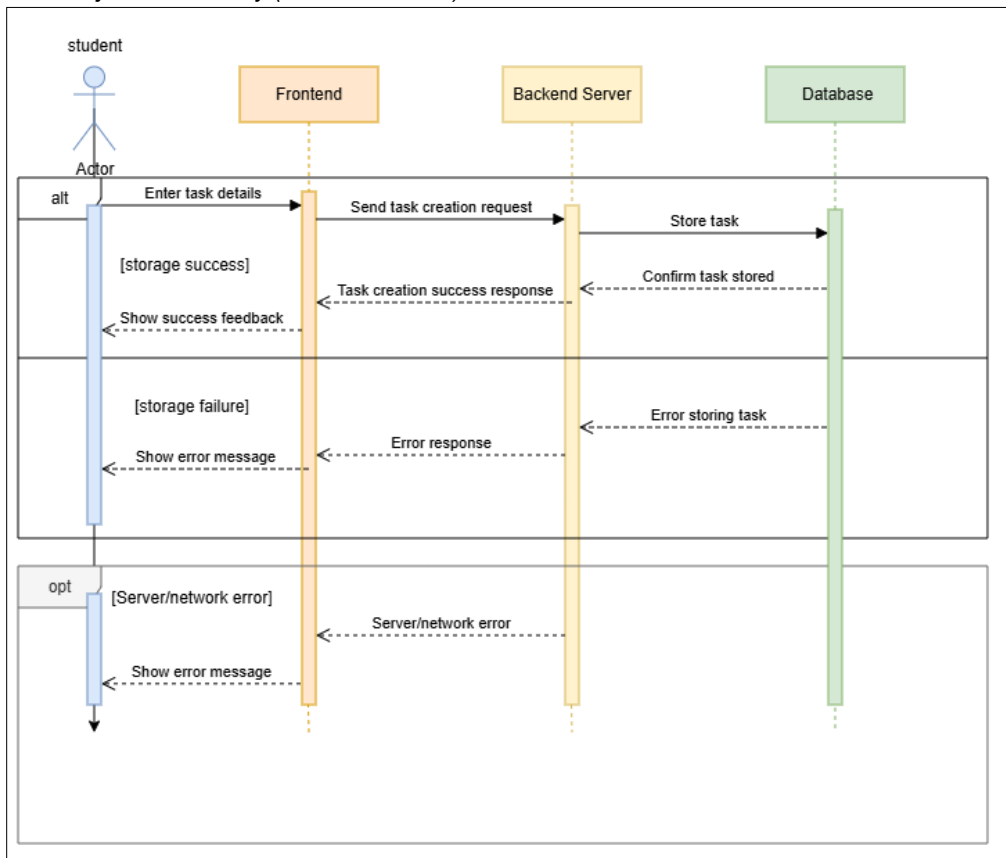
#### a) Sequence Diagram for User Login

Contributor: Ifrat Afroze Niniya (ID: 2212650042)



#### b) Task Scheduling Sequence Diagram

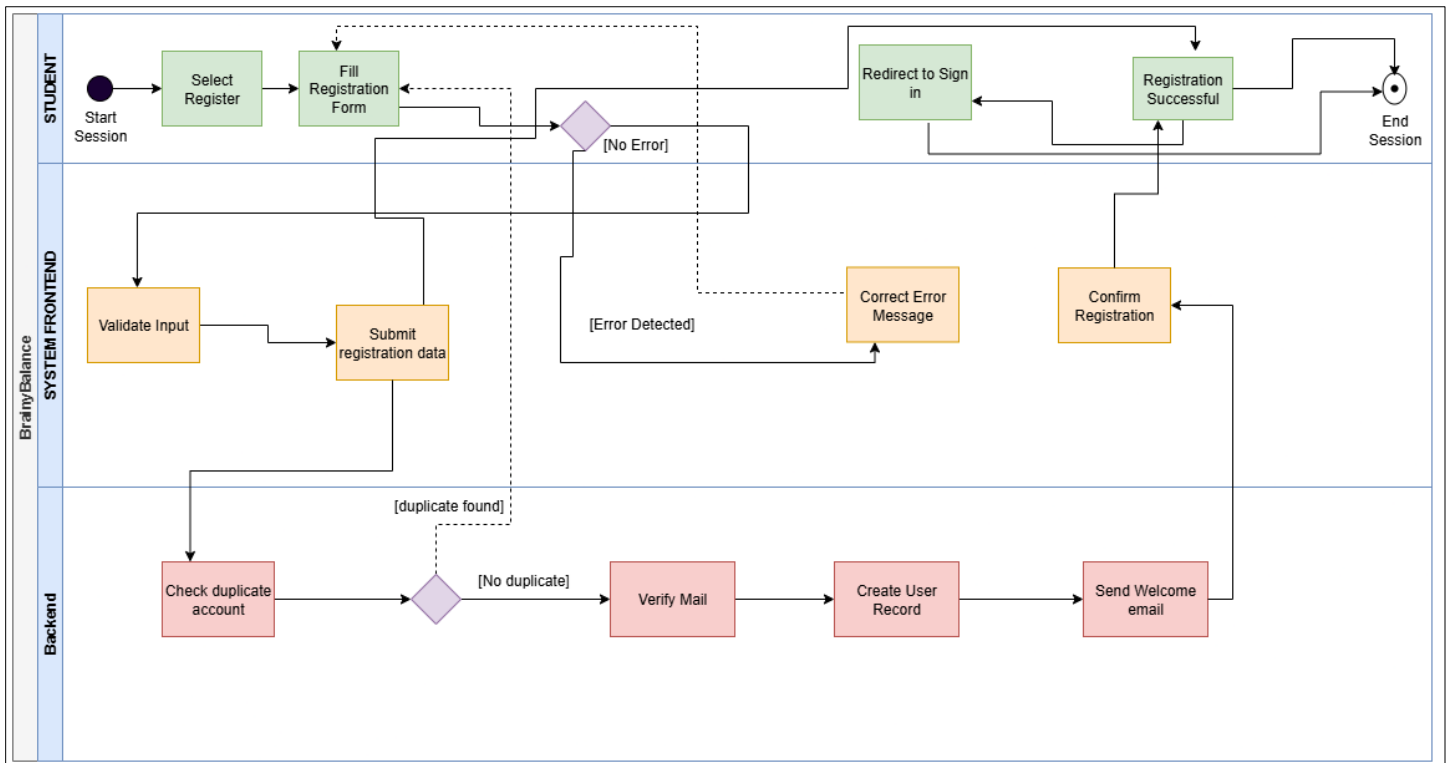
Contributor: Pujon Chakraborty (ID: 2312561642)



## 4. Activity Diagram

### a) User Registration Activity Diagram

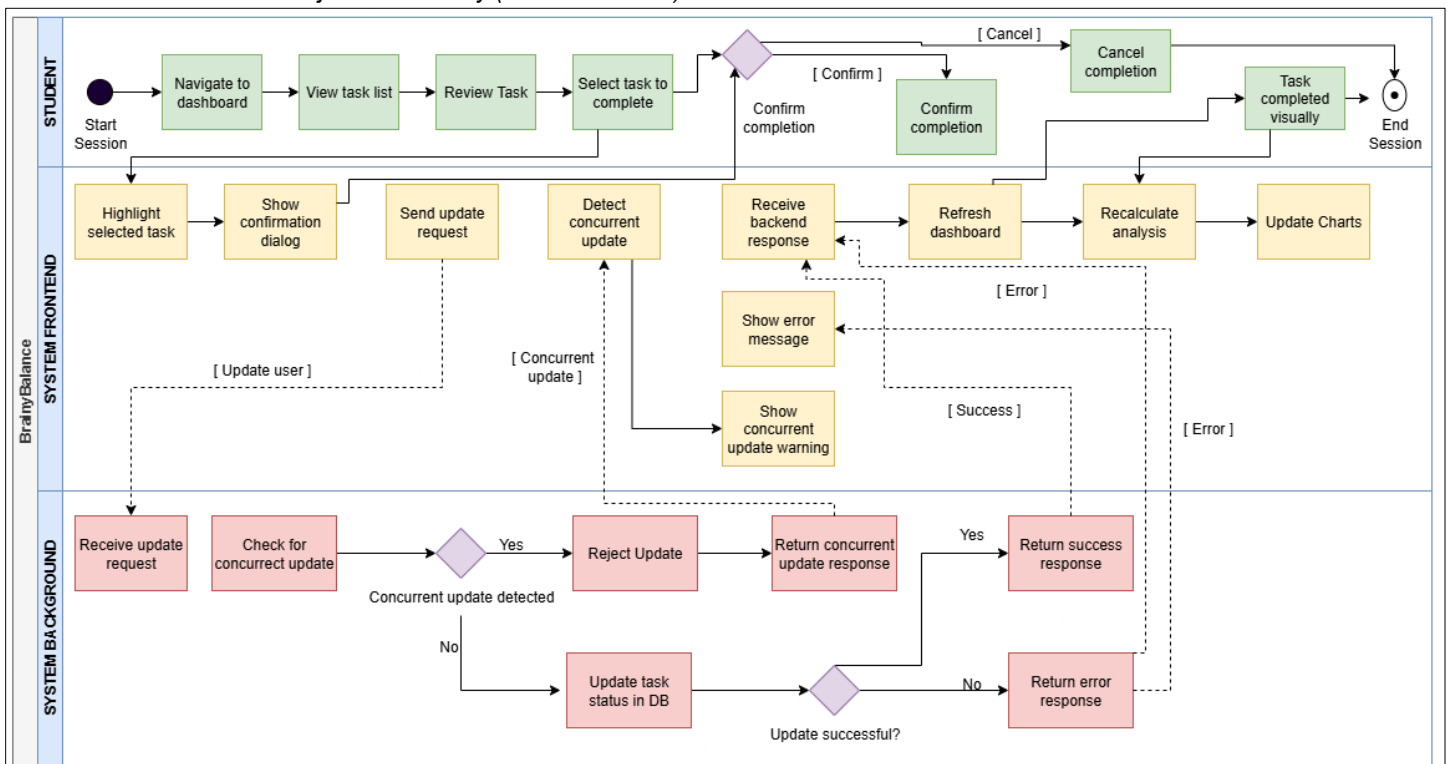
Contributor: Arafat Zaman Ratul (ID: 2311539042)



### b) Task Completion Activity Diagram

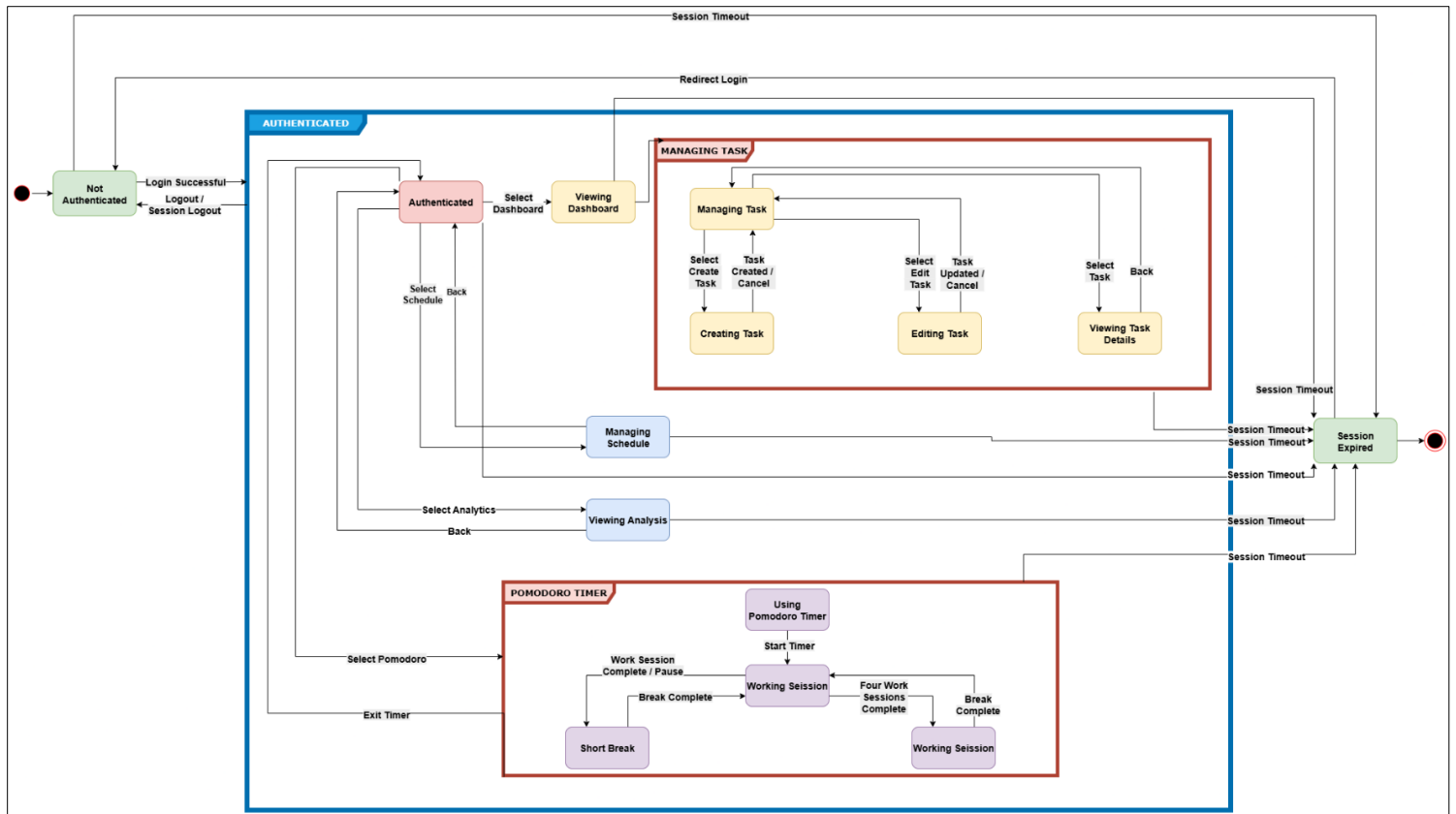
Contributors: Sudipto Roy (ID: 2222756042)

Pujon Chakraborty (ID: 2312561642)



## 5. State Machine Diagram:

Contributors: Sudipto Roy (ID: 2222756042)



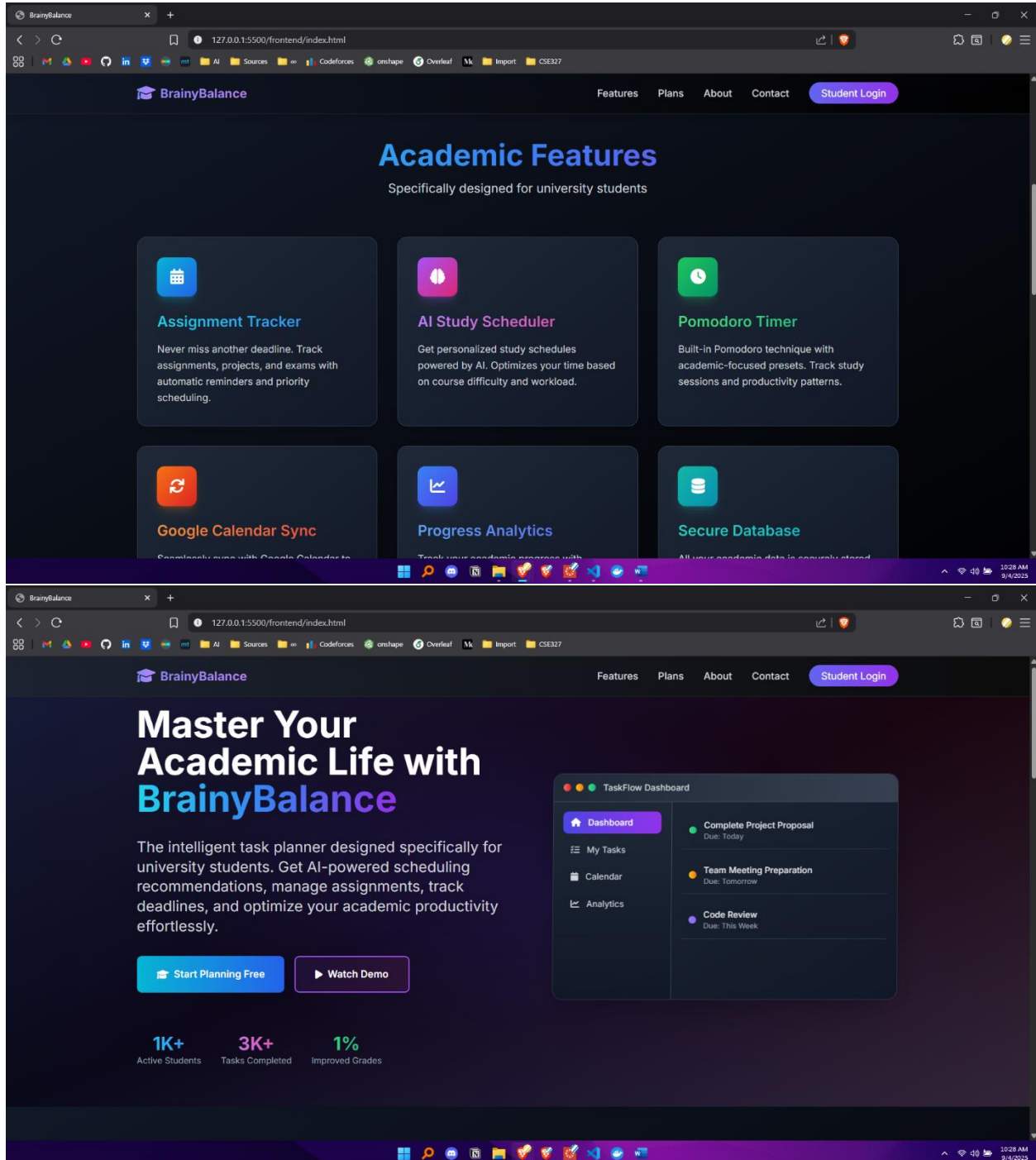
# User Access Control

Name	Dashboard	Tasks	Schedule	Analytics	Pomodoro	AI Recommendation
User	Read	Read/Write	Read	Read	Read/Write	Read

## Functionality-wise System Description (with screenshots)

### 1. Landing Page Functionality:

Landing page of our Project “BrainyBalance”. From here User can Login and Register. Since it is a landing page it is consisted of many placeholders.



## 2. Login & Register Functionality:

### Login Features:

- **Dual-form interface** with tab switching between Login and Register modes
- **Email/password authentication** with password visibility toggle and “Remember Me” option
- **Real-time validation** with user-friendly error notifications

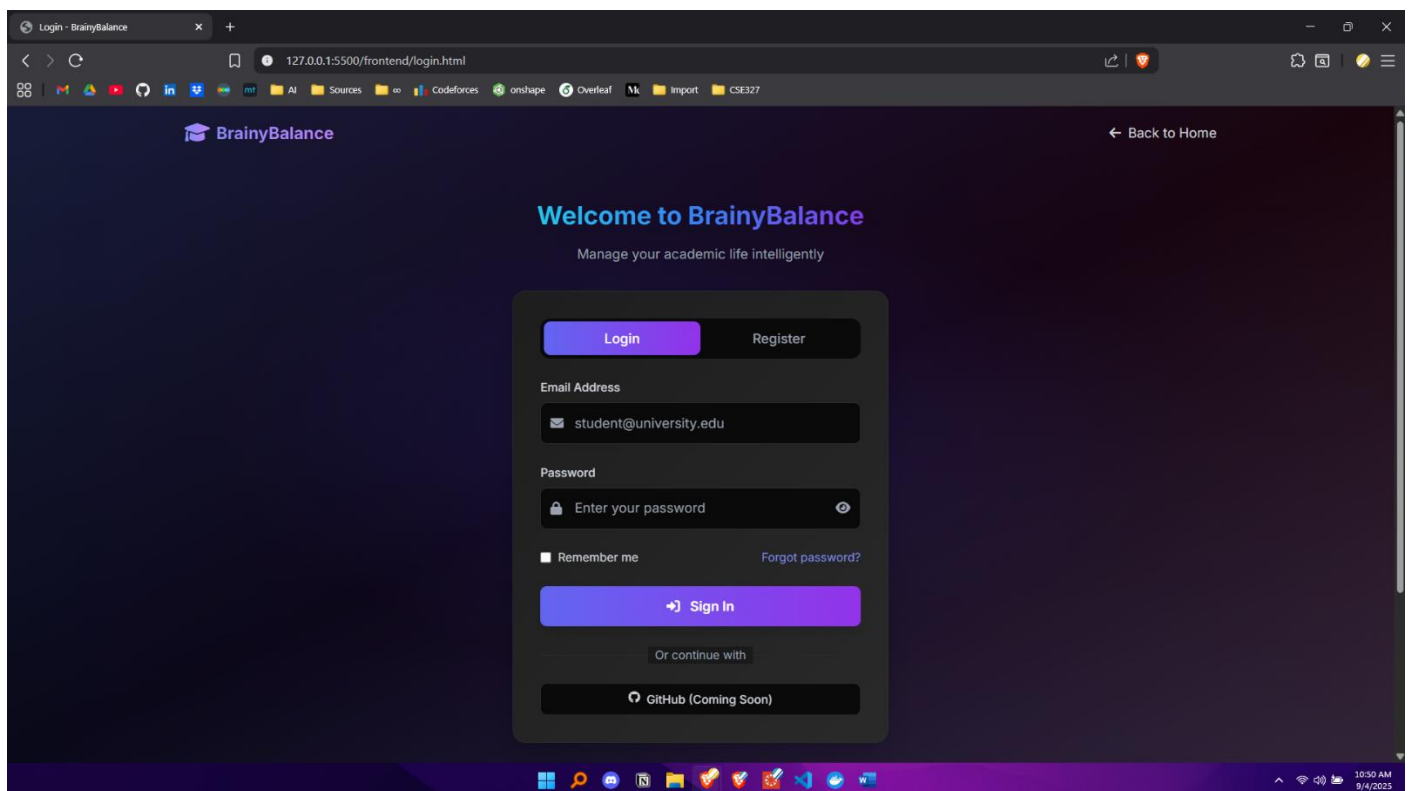
### Register Features:

- **Comprehensive form** collecting name, email, university, and secure password with confirmation
- **Password strength requirements** (minimum 6 characters) with visual feedback
- **Terms acceptance** checkbox required for account creation
- **Email verification flow** using OTP system sent to registered email
- **Automatic redirect** to verification page after successful registration

### Technical Implementation:

- **JWT token management** for secure session handling and authentication state
- **Client-side validation** with immediate feedback and API integration to backend
- **Responsive design** with dark theme, gradients, and mobile-friendly interface
- **Error handling** with specific messages for validation failures and server errors
- **Security measures** including password hashing, input sanitization, and token expiration

### *Login:*



## Register:

The screenshot shows a web browser window with the title "Login - BrainyBalance". The address bar displays "127.0.0.1:5500/frontend/login.html". The page features the "BrainyBalance" logo and a "Back to Home" link. The main heading is "Welcome to BrainyBalance" with the tagline "Manage your academic life intelligently". The registration form is centered and includes a "Login" button and a highlighted "Register" button. The form fields are: "First Name" (John), "Last Name" (Doe), "Email Address" (student@university.edu), "University" (Your University), and "Password" (Create a strong password). A note specifies that the password must be at least 8 characters with uppercase, lowercase, and numbers. A "Confirm Password" field is also present. The Windows taskbar at the bottom shows the time as 10:30 AM on 9/4/2025.

Login - BrainyBalance

127.0.0.1:5500/frontend/login.html

BrainyBalance

← Back to Home

### Welcome to BrainyBalance

Manage your academic life intelligently

Login Register

First Name Last Name

John Doe

Email Address

student@university.edu

University

Your University

Password

Create a strong password

Must be at least 8 characters with uppercase, lowercase, and numbers

Confirm Password

10:30 AM 9/4/2025

### 3. Dashboard Functionality:

#### Overview Cards:

- **Quick Stats:** Four cards showing Total Tasks, Completed Tasks, Active Projects, and Study Hours with real-time counters
- **Visual Indicators:** Color-coded cards (blue, green, purple, orange) with icons and animated hover effects
- **Progress Tracking:** Completion percentages and productivity scores

#### Today's Focus Section:

- **Today's Tasks:** Filtered view of tasks due today with priority badges and status indicators
- **Overdue Tasks:** Red-highlighted tasks past their deadlines requiring immediate attention
- **Quick Actions:** Mark complete, edit, or delete tasks directly from dashboard cards

#### Upcoming Schedule:

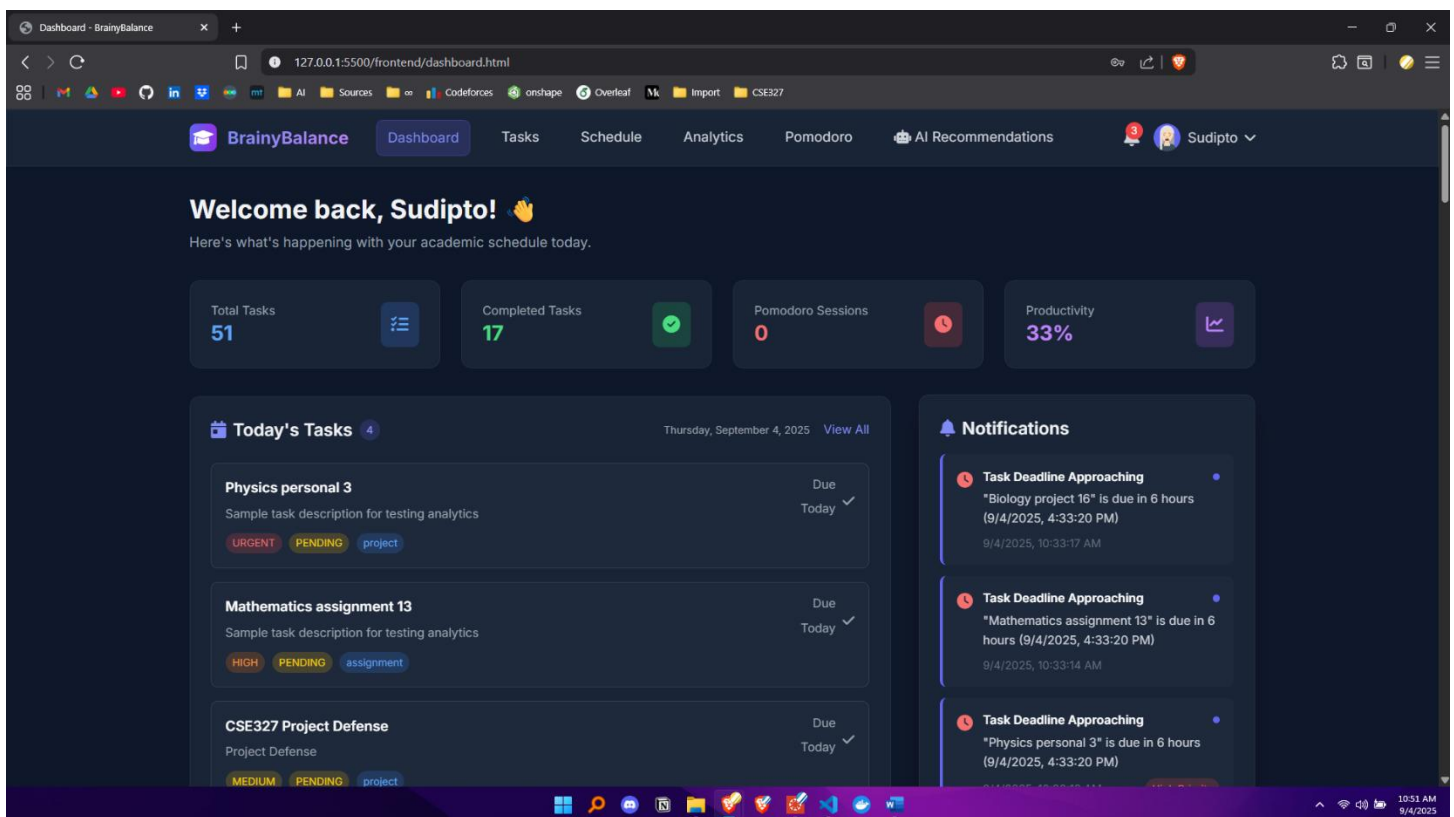
- **Next 7 Days:** Upcoming tasks and deadlines with smart date filtering
- **Priority Sorting:** High-priority items shown first with color-coded urgency levels
- **Time Estimates:** Task duration display for better time planning

#### Activity & Insights:

- **Recent Activity Feed:** Latest task completions, creations, and updates with timestamps
- **Pomodoro Stats:** Today's focus sessions and total study time integration
- **AI Recommendations:** Personalized study suggestions and productivity tips (when available)

#### Navigation Integration:

- **Quick Access:** Direct links to Tasks, Schedule, Analytics, Pomodoro, and AI Recommendations
- **Notification Center:** Bell icon with unread count and dropdown for important alerts
- **Profile Menu:** User avatar with settings and logout options





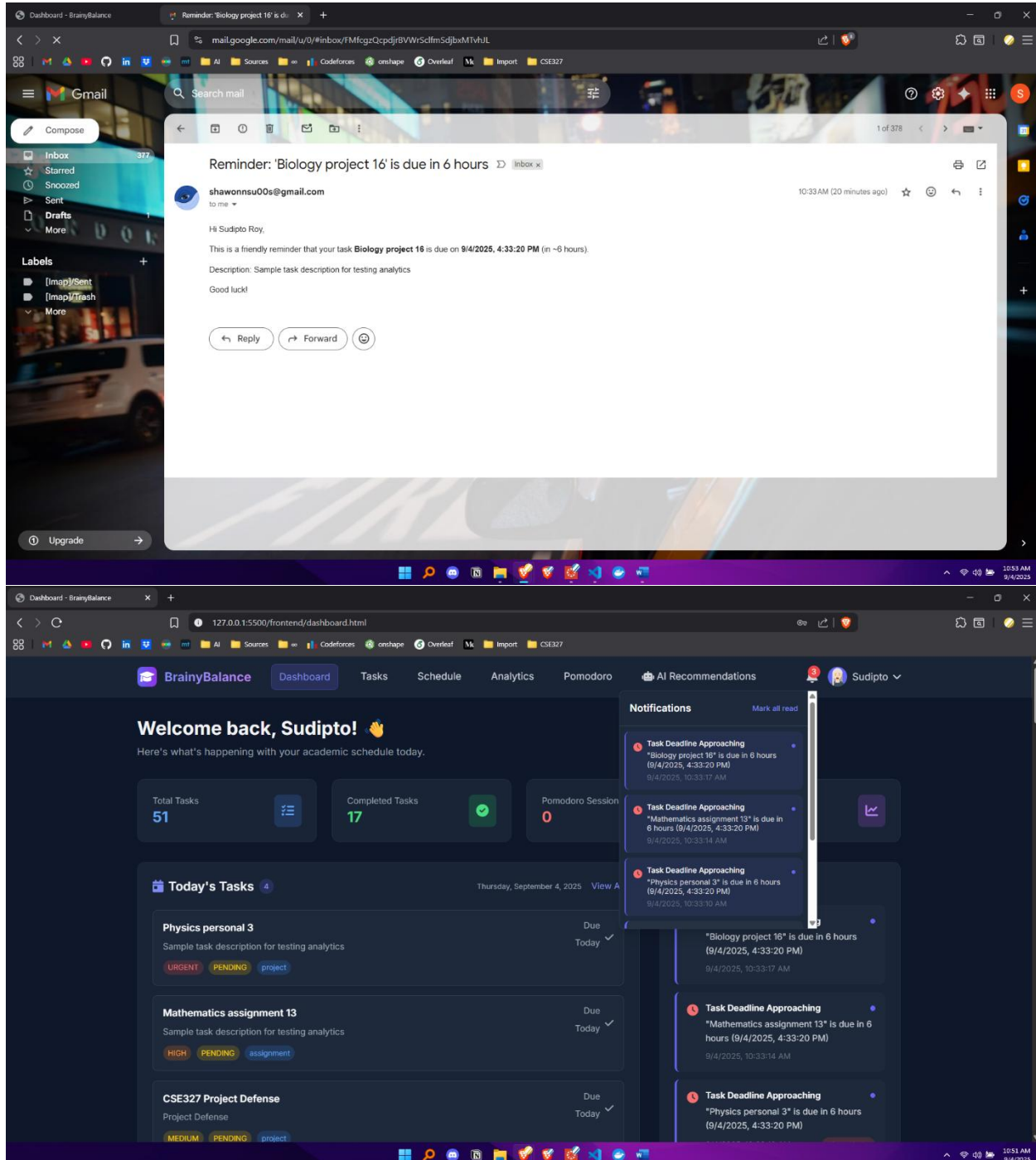
## 4. Notification & Email System for Deadlines:

### Automated Deadline Notifications:

- **Email Reminders:** System automatically sends email notifications **6 hours before** task deadlines to remind users
- **In-App Notifications:** Creates corresponding in-app notifications with deadline warnings that appear in the notification bell
- **Scheduling System:** Cron job runs every minute to check for tasks approaching their deadlines and triggers reminders
- **Duplicate Prevention:** Tracks sent reminders in database to avoid sending multiple emails for the same task

### Email Service Features:

- **SMTP Integration:** Uses Nodemailer with configurable email provider (host, port, authentication via environment variables)
- **Rich HTML Format:** Sends formatted emails with task title, description, due date, and personalized user greeting
- **Email Templates:** Professional email structure with subject line "Reminder: '[Task Title]' is due in 6 hours"



## 5. Tasks Tab Functionality:

### Core Features:

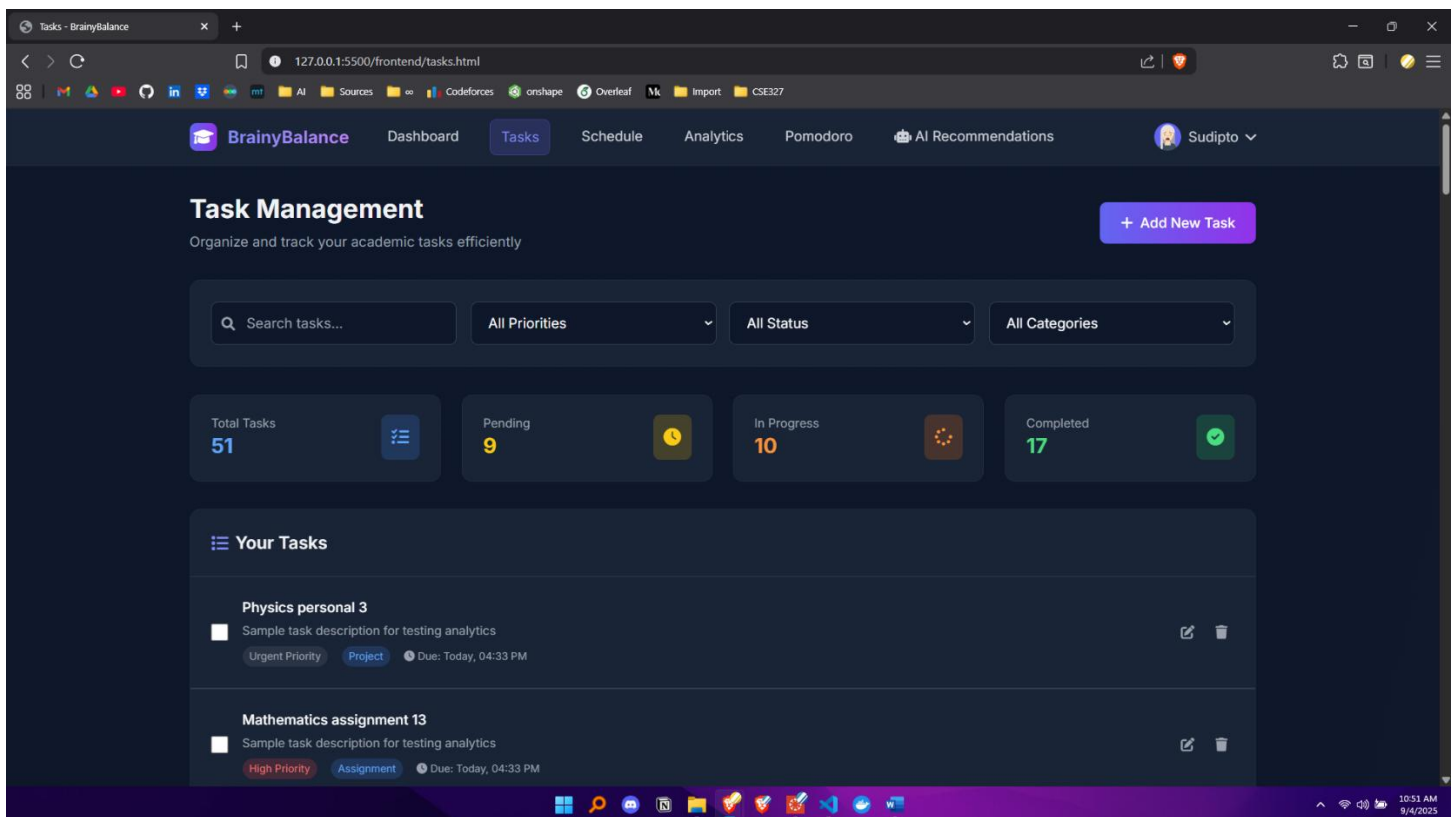
- **Task Management:** Create, edit, delete, and mark tasks complete with title, description, deadline, priority, and category
- **Advanced Filtering:** Search across all fields, filter by priority/status/category, quick filters for pending/overdue/today
- **Smart Sorting:** Upcoming tasks first (nearest deadline), then by priority, title, or creation date
- **Real-time Stats:** Live counters for total, pending, in-progress, and completed tasks with visual indicators

### Key UI Elements:

- **Task Cards:** Color-coded priority badges, category labels, deadline display with overdue highlighting
- **Filter Bar:** Search input, dropdown filters for priority/status/category, and sort options
- **Action Buttons:** Edit, delete, and completion checkboxes for each task
- **Statistics Dashboard:** Four cards showing task counts with colored icons and progress tracking

### Academic Focus:

- **Categories:** Assignment, Exam, Project, Study with appropriate organization
- **Deadline Management:** Overdue highlighting, upcoming task prioritization
- **Progress Tracking:** Completion percentages and visual progress indicators



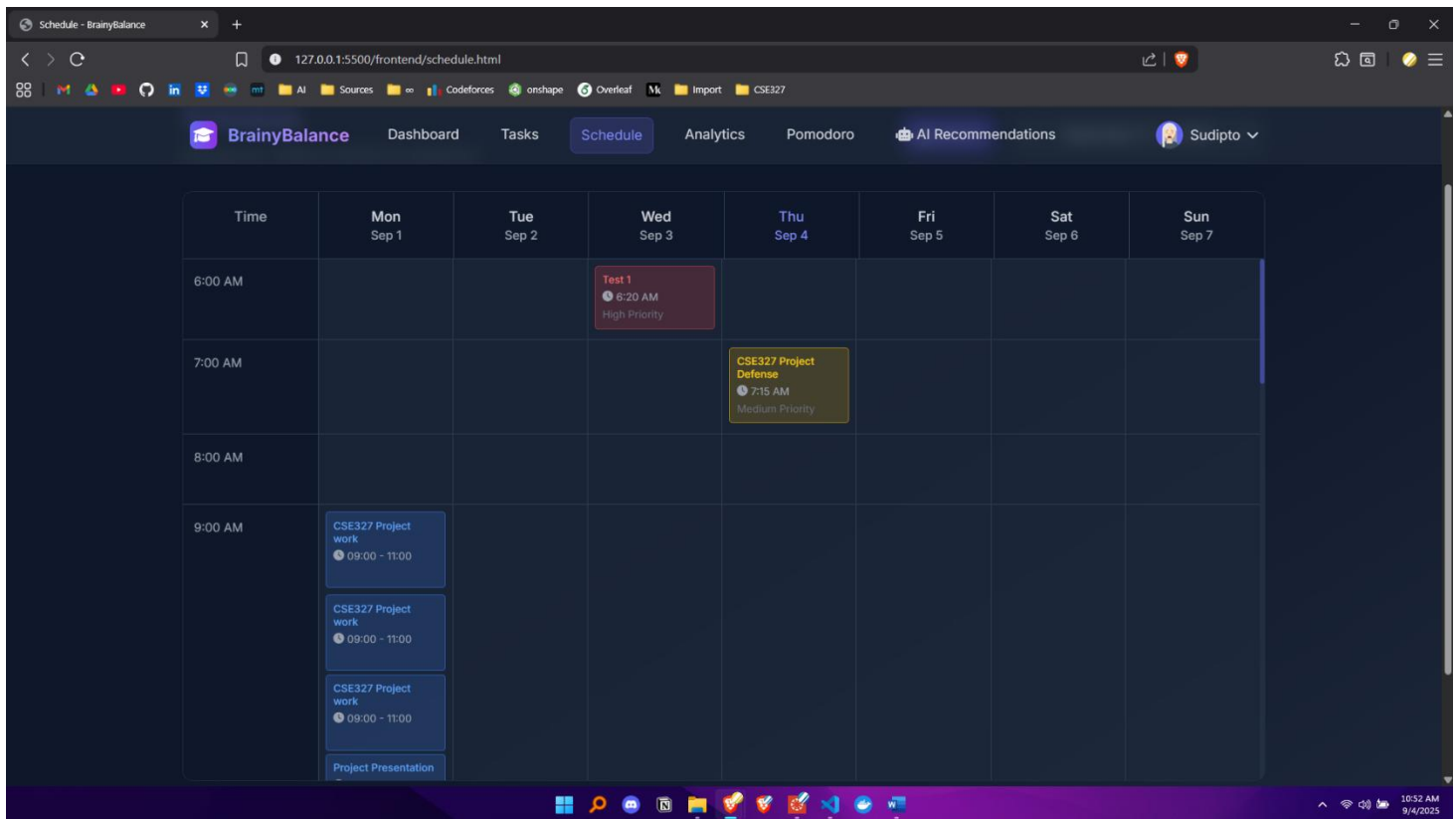
## 6. Schedule Tab Functionality:

The Schedule tab provides a comprehensive weekly calendar view that integrates both scheduled events and task deadlines, offering students a unified view of their academic commitments and time management.

### Core Features:

#### Weekly Calendar Grid:

- **Time-based Layout:** Displays hours from 6 AM to 11 PM in rows with days of the week in columns
- **Current Week Display:** Shows dates with today highlighted in indigo color
- **Week Navigation:** Previous/Next week buttons with date range display (e.g., "September 2 - 8, 2025")
- **Responsive Design:** 8-column grid (Time + 7 days) with glass morphism effects and dark theme



## 7. Analytics Tab Functionality:

The Analytics tab provides data-driven insights into student productivity, study patterns, and task completion metrics through interactive charts and real-time statistics. It focuses on **database-driven analytics**.

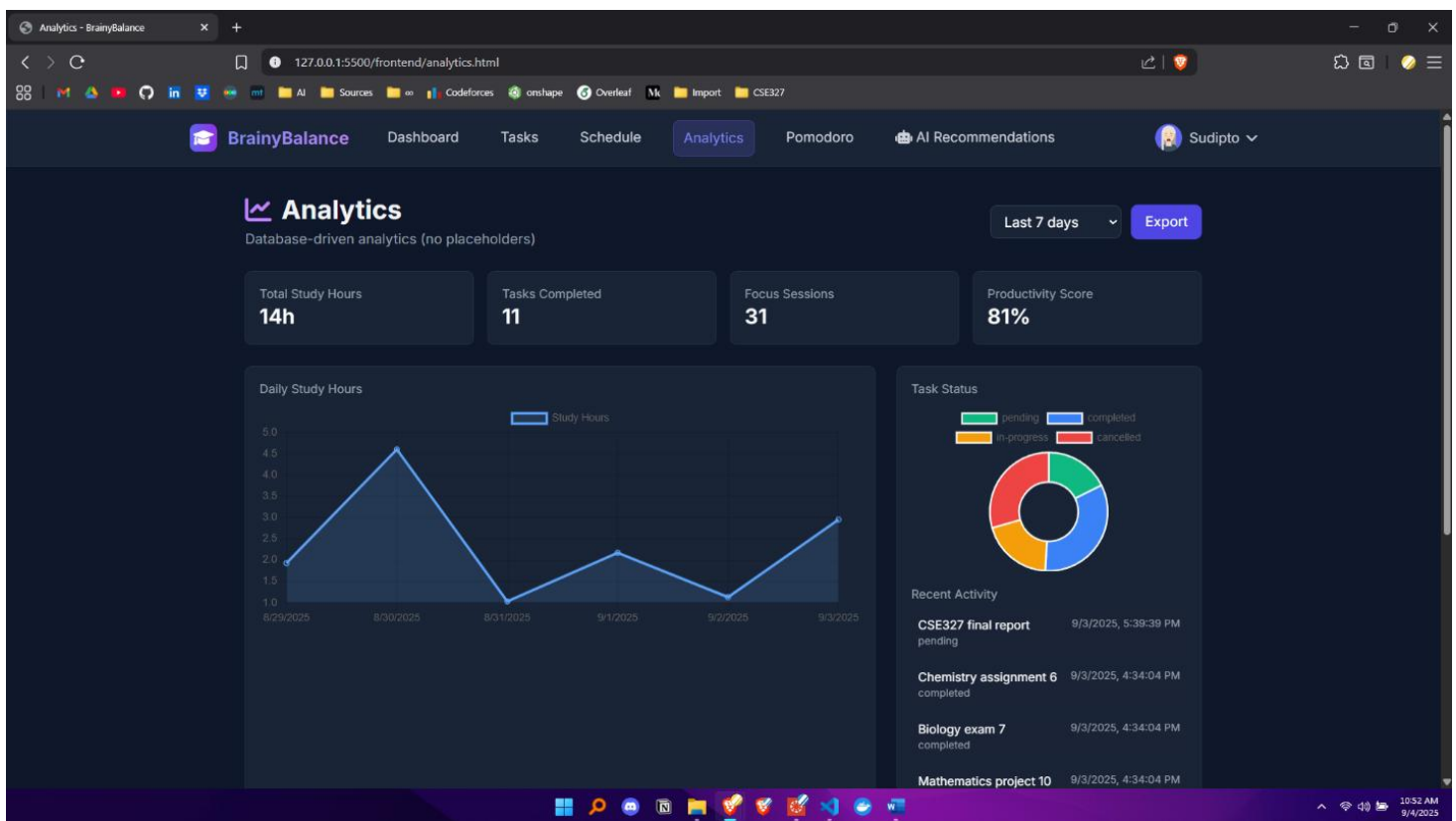
### Core Features:

#### Key Performance Metrics Dashboard:

- **Total Study Hours:** Aggregated time spent in focus sessions and study activities
- **Tasks Completed:** Count of completed tasks with real-time updates from database
- **Focus Sessions:** Number of Pomodoro sessions completed within selected time period
- **Productivity Score:** Calculated percentage based on task completion rates and study efficiency

#### Interactive Data Visualization:

- **Daily Study Hours Chart:** Line graph showing study time trends over selected period using Chart.js
- **Task Status Distribution:** Doughnut chart displaying breakdown of pending, in-progress, and completed tasks
- **Color-coded Charts:** Blue tones for study data, multi-colored segments for status distribution
- **Responsive Design:** Charts adapt to container size and maintain aspect ratio



## 8. Pomodoro Tab:

The **Pomodoro Tab** implements a comprehensive focus timer based on the Pomodoro Technique for student productivity:

### Core Timer Features:

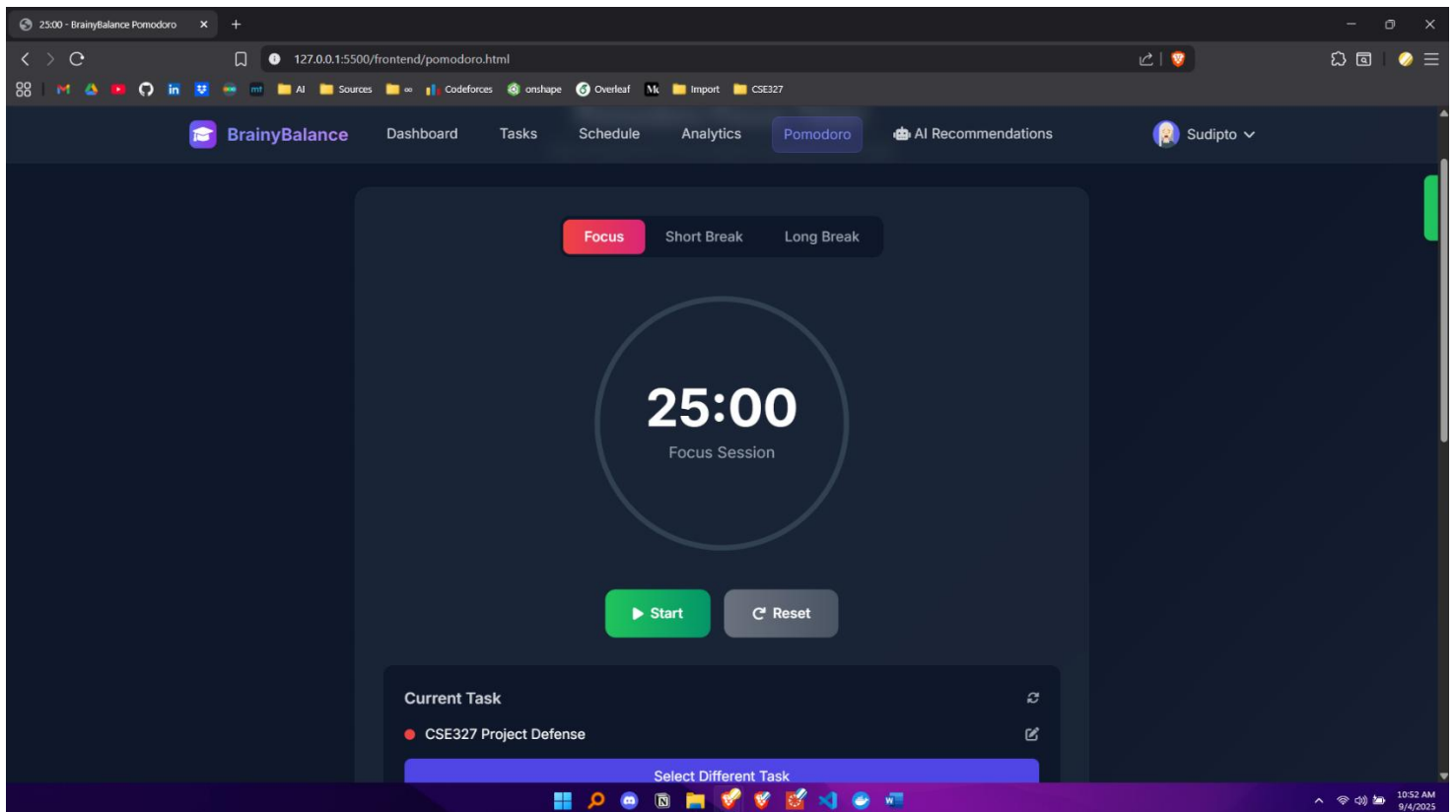
- Circular progress timer with customizable sessions (Focus: 25 min, Short Break: 5 min, Long Break: 30 min)
- Visual progress ring, start/pause/reset controls, and session type switching
- Auto-progression through focus and break cycles with configurable sessions until long break

### Task Integration:

- Database-connected task selection modal for associating focus sessions with specific tasks
- Current task display with refresh capability and real-time task loading from user's task list
- Task progress tracking linked to Pomodoro sessions

### Customizable Settings:

- Adjustable timer durations (Focus: 15-60 min, Short Break: 3-15 min, Long Break: 15-45 min)
- Configurable sessions until long break (2-8 sessions)
- Sound and notification preferences (notification sound, browser alerts, auto-start breaks)



## 9. AI Recommendations Tab:

The **AI Recommendations Tab** provides intelligent, data-driven suggestions to enhance student productivity and academic performance:

### Core AI Features:

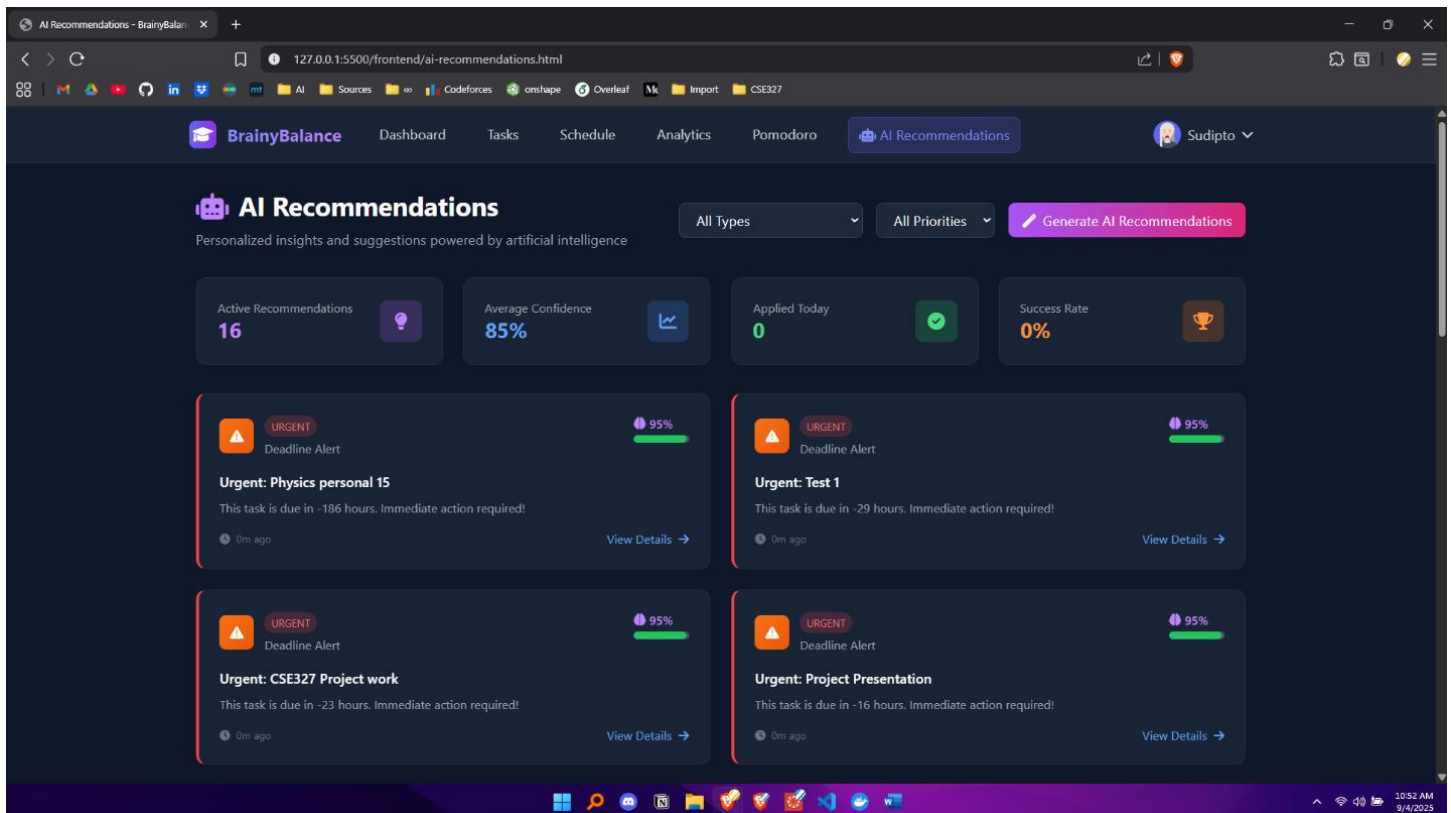
- Automated recommendation generation using Gemini analyzing user behavior patterns
- Six recommendation types: Task Priority, Study Time, Workload Balance, Deadline Alerts, Pattern Suggestions, and Schedule Optimization
- Confidence scoring system (40-100%) with visual confidence bars and color-coded reliability indicators

### Smart Analytics Dashboard:

- Real-time statistics showing active recommendations, average confidence levels, daily application rates, and success percentages
- Priority-based filtering (Urgent, High, Medium, Low) and type-based categorization for targeted insights
- Performance tracking with applied recommendations counter and overall effectiveness metrics

### Interactive Recommendation Cards:

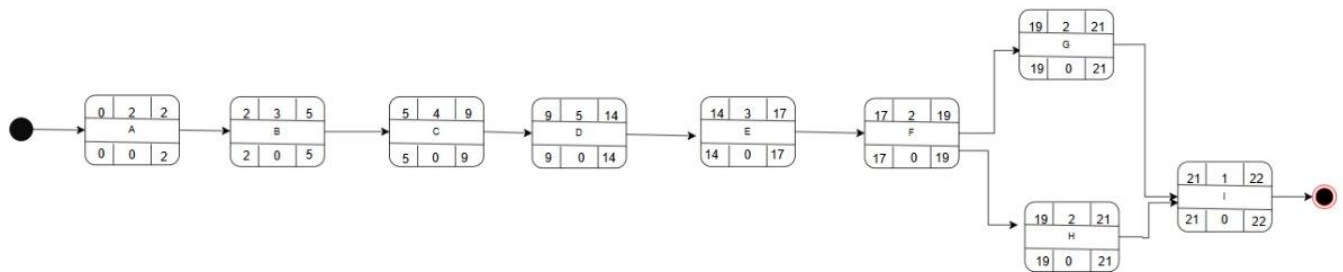
- Visual card layout with priority indicators, confidence meters, and time-based creation timestamps
- Hover effects and detailed modal views for comprehensive recommendation analysis
- Applied/pending status tracking with color-coded badges and completion indicators



## Project Scheduling

### *Critical Path Method:*

ID	Task	Duration (Days)	Dependencies
A	Requirement Analysis	2	—
B	Backend Setup	3	A
C	Task Management Logic	4	B
D	Scheduler Implementation	5	C
E	Integration and Testing	3	D
F	Final Bug Fixes	2	E
G	Deployment	2	F
H	Final Report	2	F
I	Presentation and Submission	1	H



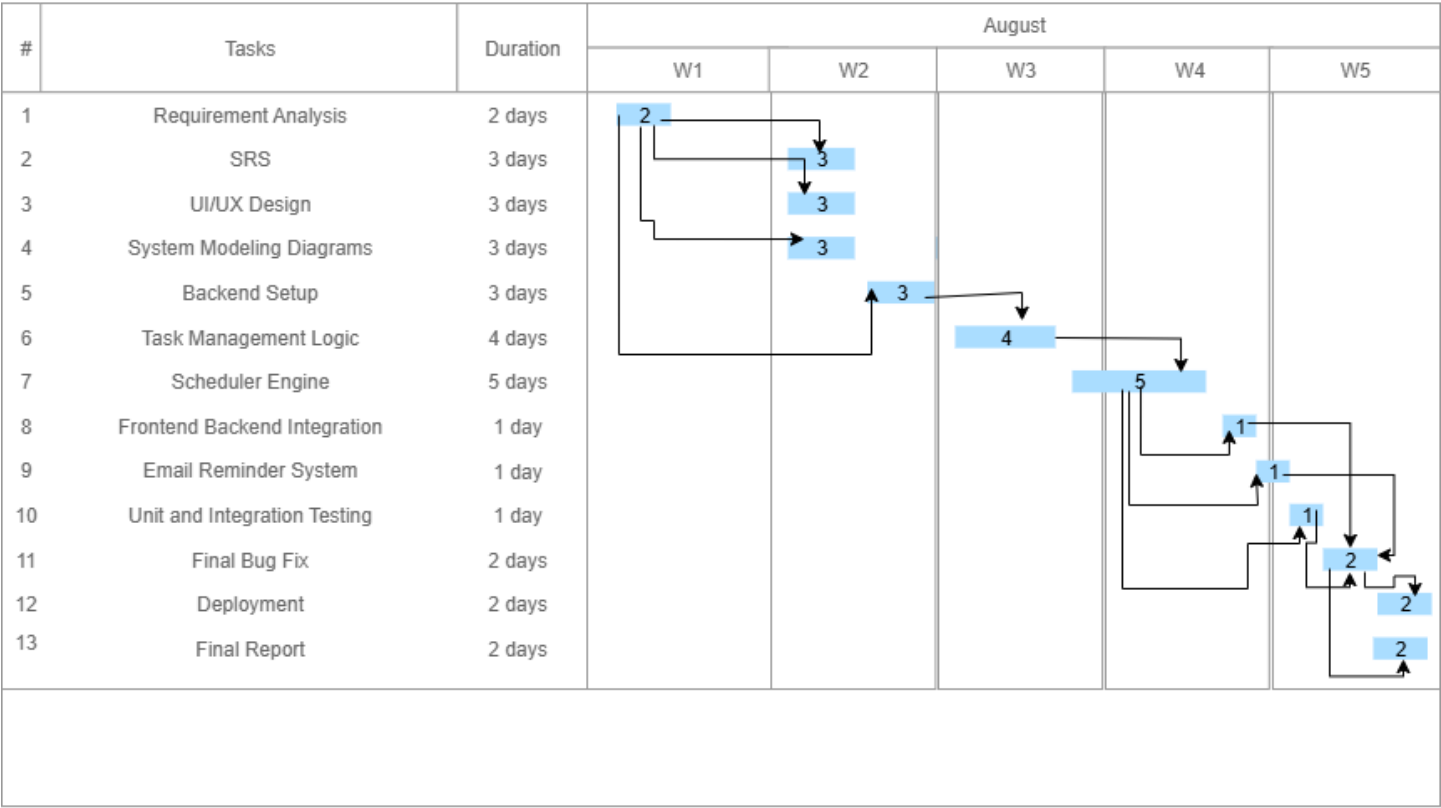
**Critical Paths:** A → B → C → D → E → F → G → I

Or

A → B → C → D → E → F → G → H

**Project Duration:** 2+3+4+5+3+2+2+1 = 22 days

Gantt chart:



## Modeling and Implementation Challenges

Modeling Challenge:

- **Task-State Complexity:** Designing of the State Machine Diagram proved to be somewhat complex due to multiple transitions. Representing optional states added ambiguity.
- **Entity Relationship Overlap:** Some entities had overlapping attributes which led to redesign efforts in the ER diagram to ensure integrity and avoid redundancy.

Implementation Challenges:

- **Frontend-Backend Synchronization:** Integrating real-time schedule generation and updates between frontend and backend API was error-prone, especially when handling user edits after auto-scheduling.
- **AI-based scheduler:** Designing a smart, conflict-free scheduler that respected time constraints, priority, and deadlines while remaining responsive was the most significant challenge.
- **Testing Email Triggers:** Automated testing of email notifications was difficult due to the external nature of email services.



## Conclusion

The “**BrainyBalance**” is a comprehensive web-based application designed to enhance academic productivity through smart scheduling, visual task tracking, and personalized reminders.

Throughout the SDLC, the team applied theoretical knowledge in practical areas including requirement analysis, UML modeling, database design, backend API development, UI creation, and testing.

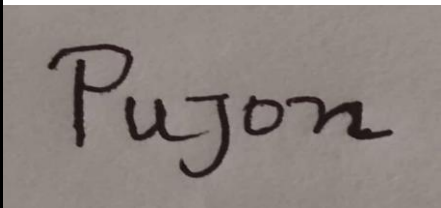
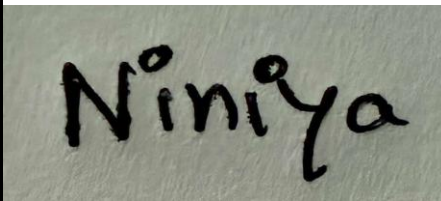
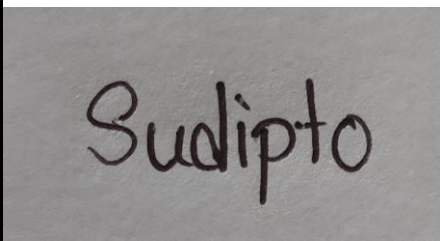
Despite facing challenges like complex task state modeling, scheduling conflicts, and front-end/backend integration issues, the final product meets the intended objectives:

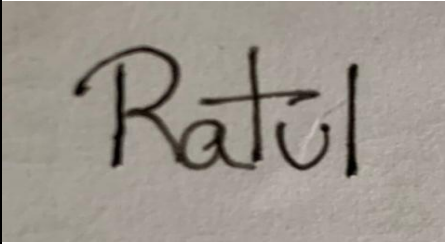
- User-friendly interface
- Rule-based task scheduler
- Visual analytics and reminders
- Functional core modules and diagrams as per SDLC

This project not only provided hands-on experience with full-stack development and AI-assisted planning but also demonstrated the importance of teamwork, iteration, and critical thinking.

By delivering hands-on full-stack work, the project showcased the value of collaboration, iterative development, and analytical thinking. Future enhancements—Google Calendar integration, a mobile client, and machine-learning-based scheduling—can expand its utility while remaining consistent with established patterns and SOLID.

**Contributions:**

ID	Name	Contributions	Signature
2312561642	Pujon Chakraborty	<b>SRS:</b> Section: Functional Requirements, Technology Stack, and Constraints  <b>System Modeling Diagrams:</b> Sequence Diagram: Task Scheduling Activity Diagram: Task Completion  <b>Final Report:</b> User Access Control, Final Report Writing  <b>Code Segment:</b> <b>Frontend:</b> Analytics, Pomodoro, AI Recommendation	
2212650042	Ifrat Afroze Niniya	<b>SRS:</b> Section: Introduction, Purpose, Scope  <b>System Modeling Diagrams:</b> Sequence Diagram: User Login Class Diagram  <b>Final Report:</b> Modeling and Implementation Challenges, Report Formatting, Conclusion  <b>Code Segment:</b> <b>Frontend:</b> Dashboard, Tasks, Schedule	
2222756042	Sudipto Roy	<b>SRS:</b> Section: User Interface Requirements, Mockup System Interface  <b>System Modeling Diagrams:</b> Activity Diagram: Task Completion, State Machine Diagram  <b>Final Report:</b> Functionalities with Screenshots, User Access Control  <b>Code Segment:</b> <b>Frontend:</b> Landing Page (index.html), Handling email remainder <b>Backend:</b> middleware, model, routes, services	

2311539042	Arafat Zaman Ratul	<p><b>SRS:</b> Section: Use Case Diagram, Product Features</p> <p><b>System Modeling Diagrams:</b> ER Diagram Activity Diagram: User Registration</p> <p><b>Final Report:</b> Project Scheduling: CPM and Gantt Chart</p> <p><b>Code Segment:</b> <b>Frontend:</b> Profile Page <b>Backend:</b> middleware, model, routes, services</p>	
------------	--------------------	---	---