SYNCORA (Project & TASK MANAGEMENT)

1. Introduction

1.1 Overview

Syncora is a modern, end-to-end project and task management application designed to simulate the way real-world software development teams work. It manages the entire Software Development Life Cycle (SDLC), from requirement gathering to deployment, allowing users to create and track projects just as they are handled in the industry. The platform enables efficient team collaboration, task assignments, project tracking, and real-time updates, making it suitable for both individual and team-based workflows. The frontend is built with React.js, styled using Material UI (MUI) for a consistent look, enhanced with Framer Motion for smooth animations, and integrated with SyncFusion for advanced data visualization like charts, calendars, and Gantt charts. The backend combines Spring Boot for scalable API development in Java and Node.js for flexible microservices. Syncora uses MongoDB for unstructured data and SQL databases for relational data, ensuring a hybrid and scalable data management approach. State management relies on cookies and localStorage for seamless session handling and persistence. AWS S3 Bucket services are integrated for secure file storage, while a built-in mail service sends automated email notifications for task assignments, status changes, and deadline alerts. Built with clean architecture and adhering to SOLID principles, Syncora ensures modular, maintainable, and testable code. It is deployed on AWS EC2 for reliable, scalable hosting, offering industry-aligned features with a strong focus on performance, design, and cloud integration.

1.2 Scope

Project Scope

The scope of the Syncora project encompasses the complete ideation, development, deployment, and ongoing support of a web-based, end-to-end project and task management platform modeled on real-world software development workflows. Intended users include software teams, project managers, and individual contributors.

In-Scope Features

- **Full SDLC Coverage**: From requirement gathering through deployment and tracking project progress. (GitHub)
- **User Authentication & Roles**: Secure sign-up, login, and potential role-based access control (if applicable).
- **Project & Task Management**: Creation, assignment, tracking, categorization of tasks with priorities, deadlines, and dependencies.
- **Collaboration Tools**: Real-time updates, notifications, comments to facilitate team coordination.
- **Advanced Data Visualization**: Interactive charts, calendars, and Gantt charts via SyncFusion integration. (GitHub)
- **Responsive UI & Animations**: Built using React.js, styled through Material UI, accentuated with Framer Motion. (GitHub)
- **Hybrid Backend Architecture**:
    - **Spring Boot** for scalable, RESTful API development in Java. (GitHub)
    - **Node.js** ecosystem implementing MongoDB-based microservices for flexible unstructured data handling. (GitHub)
- **Dual Database Strategy**: SQL for structured relational data; MongoDB for unstructured content. (GitHub)
- **State & Session Handling**: Client-side session persistence using cookies and localStorage. (GitHub)
- **Secure File Storage**: AWS S3 integration for project assets and document storage. (GitHub)
- **Automated Email Notifications**: Built-in mail service for notifying users on assignments, status updates, and approaching deadlines. (GitHub)
- **Design Principles**: Adheres to Clean Architecture and SOLID principles, ensuring modularity, scalability, and testability. (GitHub)
- **Cloud Deployment**: Hosted on AWS EC2 for reliable, scalable, and secure operations. (GitHub)

Out-of-Scope Features

- **Mobile-Native and Desktop Clients**: Focus is solely on the web-based platform in the current release.
- **Third-Party Integrations Beyond Defined Stack**: No integrations outside React, MUI, Framer Motion, SyncFusion, AWS, Spring Boot, Node.js, MongoDB, and SQL in this phase.
- **Offline Functionality or Desktop Installers**: These are not part of the scope for the initial launch.

1.3 Target Audience

Syncora is designed for software teams with clearly defined roles, ensuring that each user type has access to features and tools tailored to their responsibilities. The platform provides three main panels: **Admin**, **Tester**, and **Developer**.

1. Admin Panel

The Admin panel is intended for system administrators who manage the overall platform, user access, and project configurations.
Key capabilities include:

- Creating and managing user accounts with appropriate roles and permissions.
- Overseeing all active projects and task assignments.
- Managing system configurations, integrations, and data security.
- Monitoring platform activity logs and generating usage reports.
- Handling escalated issues and making platform-wide announcements.

2. Tester Panel

The Tester panel is designed for Quality Assurance (QA) professionals responsible for validating product quality throughout the Software Development Life Cycle.
Key capabilities include:

- Accessing and tracking assigned testing tasks.
- Reporting bugs with detailed descriptions, reproduction steps, and severity levels.
- Collaborating with developers to ensure issues are resolved efficiently.
- Retesting resolved bugs and marking them as closed.

- Monitoring testing progress through reports and dashboards.

3. Developer Panel

The Developer panel is meant for software engineers responsible for implementing features, fixing bugs, and collaborating with testers and managers.
Key capabilities include:

- Viewing and managing assigned development tasks.
- Updating task status in real-time (e.g., in progress, completed).
- Accessing project requirements, technical documentation, and relevant assets.
- Collaborating with testers to resolve bugs quickly.
- Tracking personal and team progress through dashboards and charts.

Role–Feature Mapping Table

Role	Key Responsibilities	Core Syncora Features
Admin	- Manage user accounts, roles, and permissions.- Oversee all active projects and monitor task assignments.- Configure platform settings and integrations.- Maintain data security and handle escalated issues.	- **User Management Module** for creating/updating users.- **Role & Permission Control** for access management.- **System Logs & Reports** for monitoring activity.- **AWS S3 File Management** for secure storage.- **Global Notifications** for platform announcements.
Tester	- Validate application functionality.- Execute test cases and log bugs.- Collaborate with developers for issue resolution.- Retest fixed issues and close resolved bugs.- Track testing progress.	- **Bug Tracking System** for logging issues.- **Task Board Integration** for assigned test cases.- **Commenting & Collaboration Tools** for dev–QA communication.- **Reports & Dashboards** for testing progress.

Developer	- Implement assigned features and enhancements.- Fix reported bugs.- Update task statuses in real time.- Collaborate with testers and admins.- Access documentation and assets.	- **Task Management Board** for viewing & updating tasks.- **Real-Time Status Updates** for progress tracking.- **Document & Asset Access** via AWS S3.- **Collaboration Tools** for communication with QA and admin.- **Progress Dashboards** for personal and team metrics.

---

2. Tech Stack

Syncora is built using a modern, scalable, and maintainable technology stack designed to deliver high performance, flexibility, and security.

2.1. Frontend

The frontend of Syncora is designed for a responsive and interactive user experience, combining modern JavaScript frameworks with powerful UI libraries.

- **React.js** – Core JavaScript library for building the user interface and component-based architecture.
- **Material UI (MUI)** – Provides a consistent and professional design system with pre-built, customizable components.
- **Framer Motion** – Enables smooth animations and transitions to enhance user experience.
- **SyncFusion** – Used for advanced data visualization components such as charts, calendars, and Gantt charts.
- **Axios / Fetch API** – For API communication with backend services.
- **CSS Modules & Theming** – For modular, maintainable styling and theme customization.

2.2 Backend

The backend architecture follows a hybrid approach to leverage the strengths of both monolithic and microservice patterns.

- **Spring Boot (Java)** – For building robust, scalable, and secure REST APIs.

- **Node.js (JavaScript)** – For implementing lightweight microservices and handling unstructured data processing.
- **Express.js** – Framework for Node.js microservices to handle routing and middleware.
- **MongoDB** – NoSQL database for storing unstructured and semi-structured data.
- **MySQL / PostgreSQL** – Relational database for structured data storage.
- **JWT Authentication** – For secure, token-based user authentication and authorization.
- **AWS S3 Integration** – For file storage and retrieval.
- **Mail Service (SMTP / AWS SES)** – For automated email notifications.

2.3. Infrastructure

The infrastructure ensures that Syncora is scalable, reliable, and secure in production environments.

- **AWS EC2** – Cloud hosting for deploying backend services.
- **AWS S3** – Object storage for documents, images, and project-related files.
- **AWS IAM** – Identity and Access Management for secure service access.
- **Git & GitHub** – Version control and collaborative development.

3.Requirements

3.1 Functional Requirements

The following functional requirements define what Syncora must achieve to meet its objectives:

User Management

- **Admin Panel**:
    - Create, update, and delete user accounts (Admin, Developer, Tester).

    - Assign roles and permissions to users.
    - Monitor overall system activity and manage system-wide settings.

Project & Task Management

- **Project Creation & Tracking**:
    - Create new projects with detailed descriptions, deadlines, and assigned team members.
    - Track project progress through Kanban boards, Gantt charts, and calendar views (via SyncFusion).
- **Task Management**:
    - Assign tasks to developers or testers with deadlines and priority levels.
    - Update task statuses (To Do, In Progress, Completed).
    - Add attachments to tasks (stored securely in AWS S3).

Collaboration & Communication

- Real-time updates for task changes, assignments, and project status.
- Built-in email notifications for:
    - New task assignments.
    - Status changes.
    - Upcoming deadlines.

Testing & Quality Assurance

- Tester role to log bugs/issues with detailed descriptions.
- Assign bugs to developers for resolution.
- Track bug-fix progress and close issues upon verification.

File & Document Management

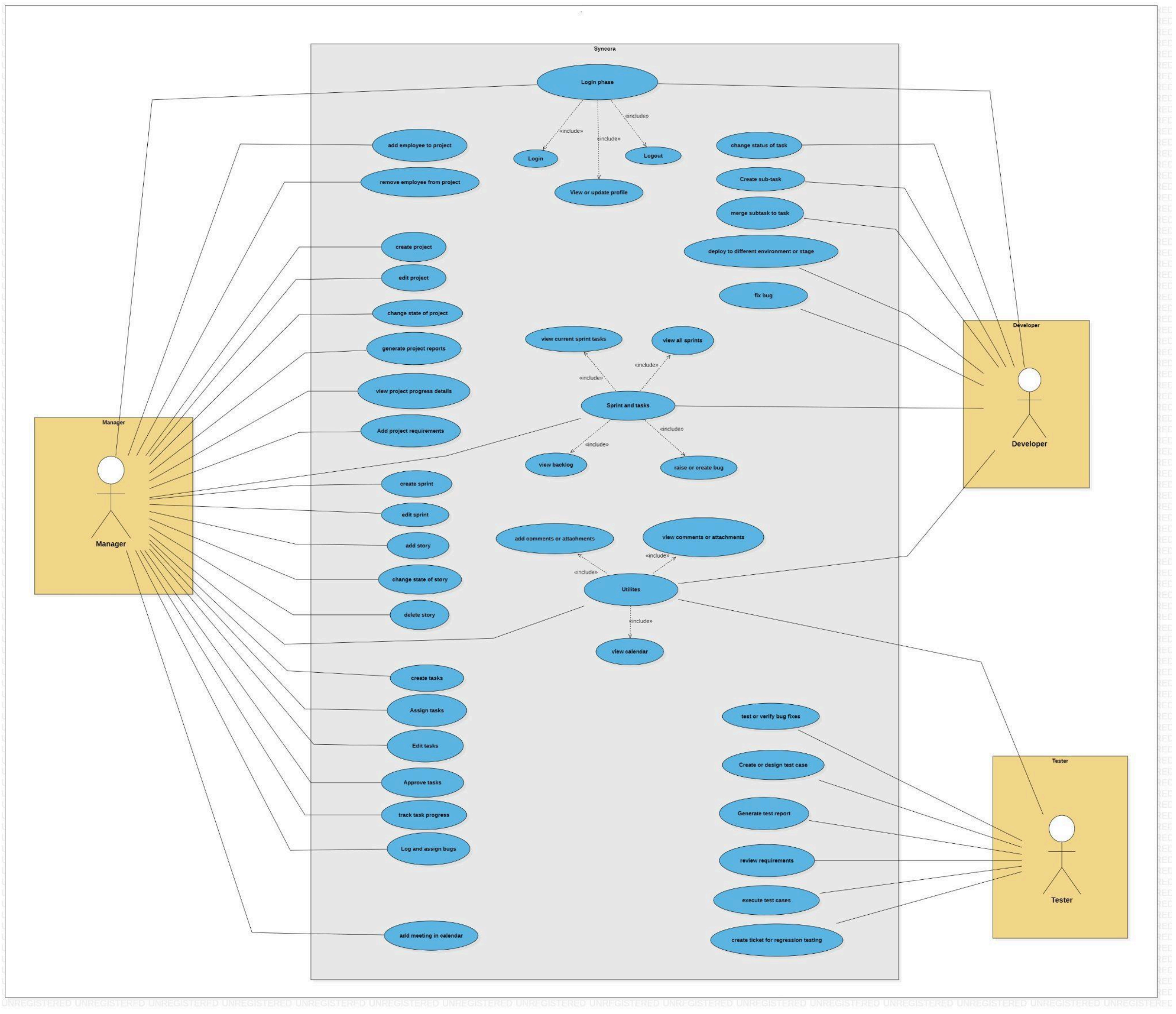
- Upload and manage project-related files securely via AWS S3.
- Access control based on user roles to ensure data confidentiality.

Data Visualization & Reporting

- Interactive charts and graphs for project timelines, workload distribution, and performance tracking.
- Exportable reports for management and auditing.

Security & Access Control

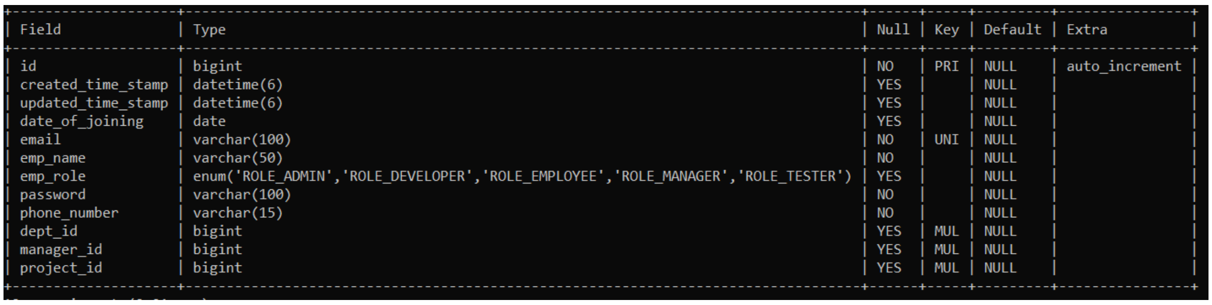
- Role-based authentication and authorization.
- Secure session management using cookies and localStorage.
- Input validation to prevent malicious data entry.



4.Design

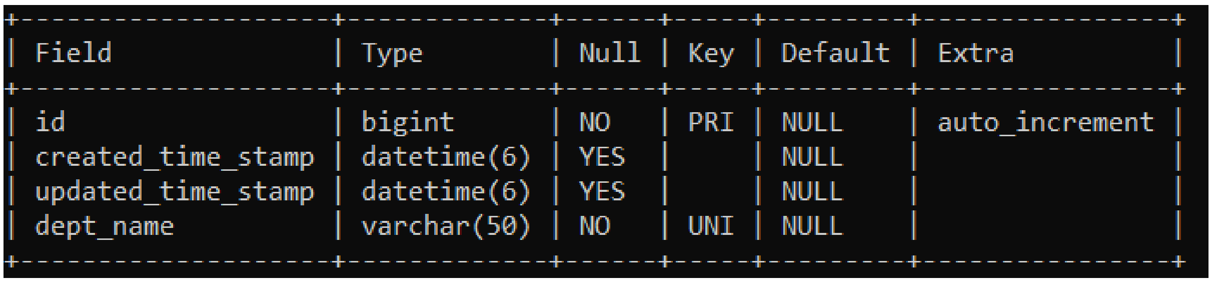
4.1.Database Design

Table-1-:Employee Table



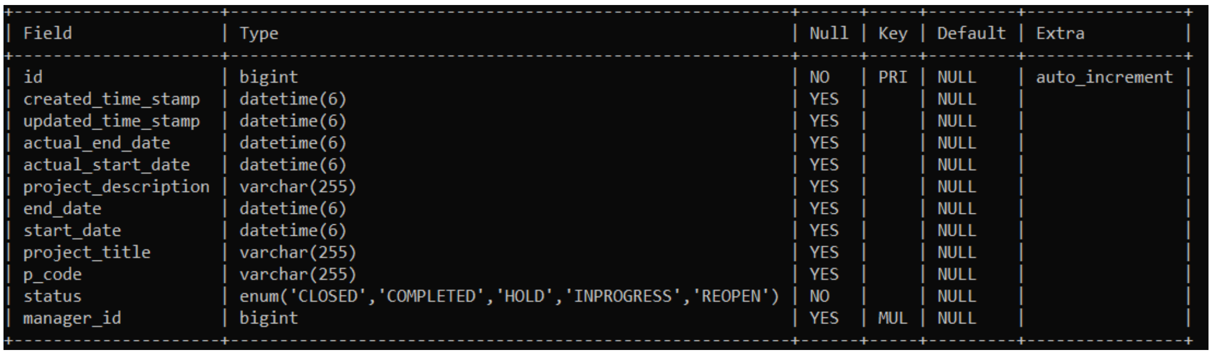
```
+--------------------+----------------------------------------------------------------------------------+------+-----+---------+----------------+
| Field              | Type                                                                             | Null | Key | Default | Extra          |
+--------------------+----------------------------------------------------------------------------------+------+-----+---------+----------------+
| id                 | bigint                                                                           | NO   | PRI | NULL    | auto_increment |
| created_time_stamp | datetime(6)                                                                      | YES  |     | NULL    |                |
| updated_time_stamp | datetime(6)                                                                      | YES  |     | NULL    |                |
| date_of_joining    | date                                                                             | YES  |     | NULL    |                |
| email              | varchar(100)                                                                     | NO   | UNI | NULL    |                |
| emp_name           | varchar(50)                                                                      | NO   |     | NULL    |                |
| emp_role           | enum('ROLE_ADMIN','ROLE_DEVELOPER','ROLE_EMPLOYEE','ROLE_MANAGER','ROLE_TESTER') | YES  |     | NULL    |                |
| password           | varchar(100)                                                                     | NO   |     | NULL    |                |
| phone_number       | varchar(15)                                                                      | NO   |     | NULL    |                |
| dept_id            | bigint                                                                           | YES  | MUL | NULL    |                |
| manager_id         | bigint                                                                           | YES  | MUL | NULL    |                |
| project_id         | bigint                                                                           | YES  | MUL | NULL    |                |
+--------------------+----------------------------------------------------------------------------------+------+-----+---------+----------------+
```

Table-2-:Departments



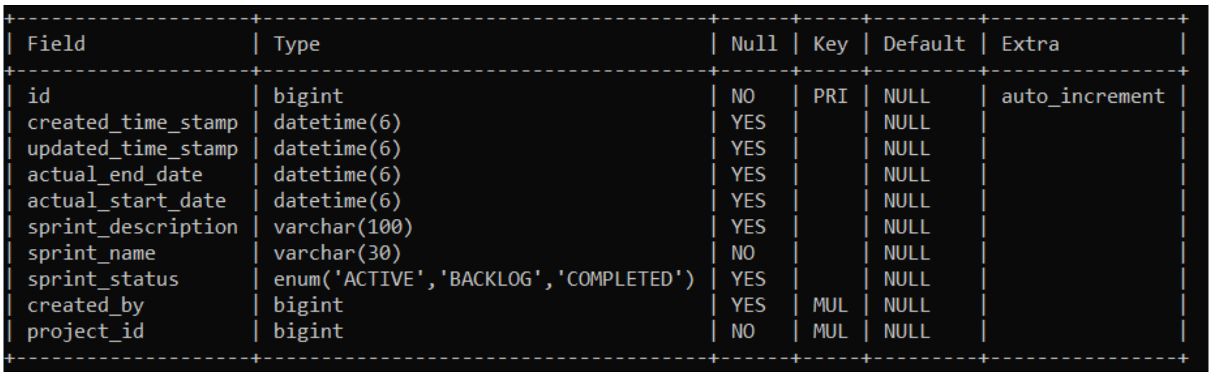
```
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| id                 | bigint      | NO   | PRI | NULL    | auto_increment |
| created_time_stamp | datetime(6) | YES  |     | NULL    |                |
| updated_time_stamp | datetime(6) | YES  |     | NULL    |                |
| dept_name          | varchar(50) | NO   | UNI | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
```

Table-3-:Projects



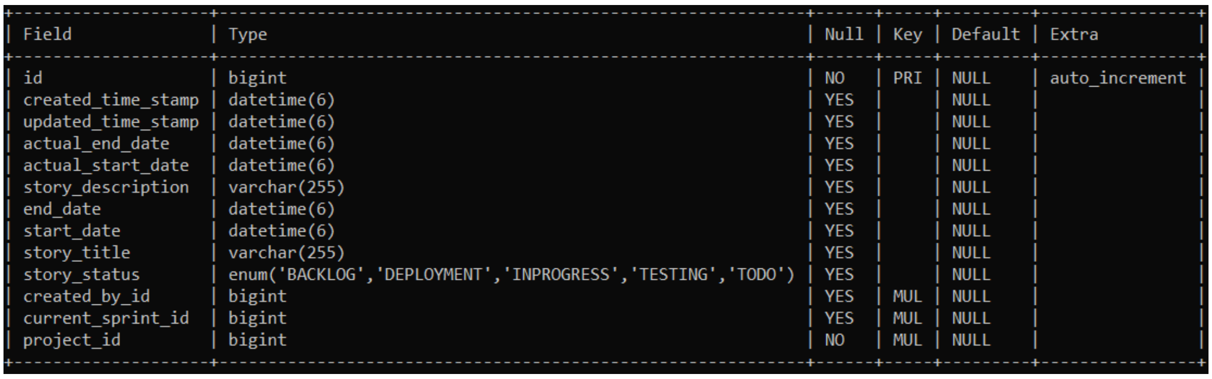
```
+---------------------+---------------------------------------------------------+------+-----+---------+----------------+
| Field               | Type                                                    | Null | Key | Default | Extra          |
+---------------------+---------------------------------------------------------+------+-----+---------+----------------+
| id                  | bigint                                                  | NO   | PRI | NULL    | auto_increment |
| created_time_stamp  | datetime(6)                                             | YES  |     | NULL    |                |
| updated_time_stamp  | datetime(6)                                             | YES  |     | NULL    |                |
| actual_end_date     | datetime(6)                                             | YES  |     | NULL    |                |
| actual_start_date   | datetime(6)                                             | YES  |     | NULL    |                |
| project_description | varchar(255)                                            | YES  |     | NULL    |                |
| end_date            | datetime(6)                                             | YES  |     | NULL    |                |
| start_date          | datetime(6)                                             | YES  |     | NULL    |                |
| project_title       | varchar(255)                                            | YES  |     | NULL    |                |
| p_code              | varchar(255)                                            | YES  |     | NULL    |                |
| status              | enum('CLOSED','COMPLETED','HOLD','INPROGRESS','REOPEN') | NO   |     | NULL    |                |
| manager_id          | bigint                                                  | YES  | MUL | NULL    |                |
+---------------------+---------------------------------------------------------+------+-----+---------+----------------+
```

Table-4-:Sprints



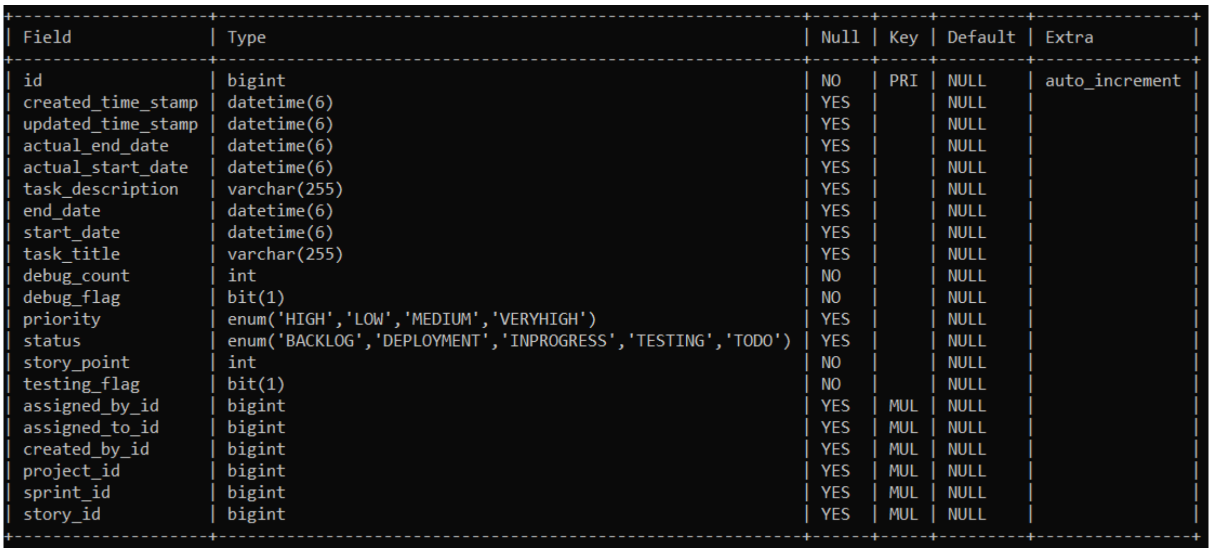
```
+--------------------+--------------------------------------+------+-----+---------+----------------+
| Field              | Type                                 | Null | Key | Default | Extra          |
+--------------------+--------------------------------------+------+-----+---------+----------------+
| id                 | bigint                               | NO   | PRI | NULL    | auto_increment |
| created_time_stamp | datetime(6)                          | YES  |     | NULL    |                |
| updated_time_stamp | datetime(6)                          | YES  |     | NULL    |                |
| actual_end_date    | datetime(6)                          | YES  |     | NULL    |                |
| actual_start_date  | datetime(6)                          | YES  |     | NULL    |                |
| sprint_description | varchar(100)                         | YES  |     | NULL    |                |
| sprint_name        | varchar(30)                          | NO   |     | NULL    |                |
| sprint_status      | enum('ACTIVE','BACKLOG','COMPLETED') | YES  |     | NULL    |                |
| created_by         | bigint                               | YES  | MUL | NULL    |                |
| project_id         | bigint                               | NO   | MUL | NULL    |                |
+--------------------+--------------------------------------+------+-----+---------+----------------+
```

Table-5-:Stories



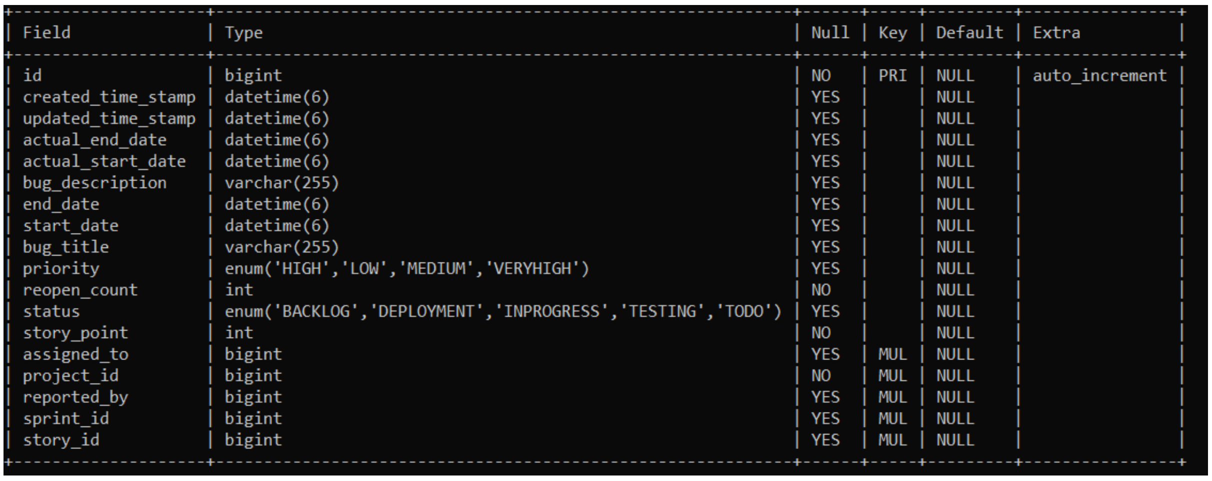
```
+--------------------+----------------------------------------------------------+------+-----+---------+----------------+
| Field              | Type                                                     | Null | Key | Default | Extra          |
+--------------------+----------------------------------------------------------+------+-----+---------+----------------+
| id                 | bigint                                                   | NO   | PRI | NULL    | auto_increment |
| created_time_stamp | datetime(6)                                              | YES  |     | NULL    |                |
| updated_time_stamp | datetime(6)                                              | YES  |     | NULL    |                |
| actual_end_date    | datetime(6)                                              | YES  |     | NULL    |                |
| actual_start_date  | datetime(6)                                              | YES  |     | NULL    |                |
| story_description  | varchar(255)                                             | YES  |     | NULL    |                |
| end_date           | datetime(6)                                              | YES  |     | NULL    |                |
| start_date         | datetime(6)                                              | YES  |     | NULL    |                |
| story_title        | varchar(255)                                             | YES  |     | NULL    |                |
| story_status       | enum('BACKLOG','DEPLOYMENT','INPROGRESS','TESTING','TODO') | YES  |     | NULL    |                |
| created_by_id      | bigint                                                   | YES  | MUL | NULL    |                |
| current_sprint_id  | bigint                                                   | YES  | MUL | NULL    |                |
| project_id         | bigint                                                   | NO   | MUL | NULL    |                |
+--------------------+----------------------------------------------------------+------+-----+---------+----------------+
```

Table-6-:Tasks



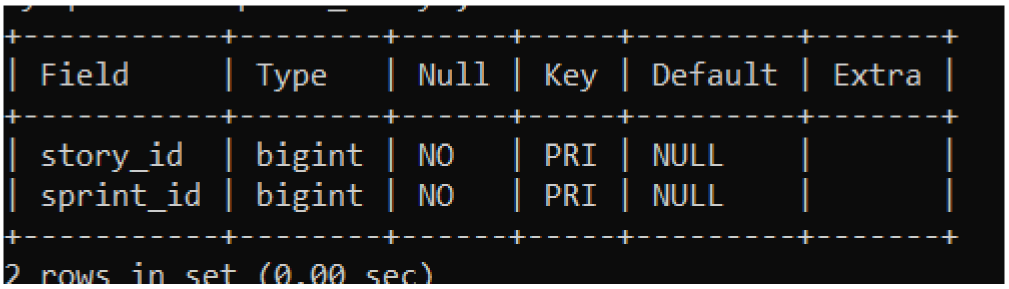
```
+--------------------+----------------------------------------------------------+------+-----+---------+----------------+
| Field              | Type                                                     | Null | Key | Default | Extra          |
+--------------------+----------------------------------------------------------+------+-----+---------+----------------+
| id                 | bigint                                                   | NO   | PRI | NULL    | auto_increment |
| created_time_stamp | datetime(6)                                              | YES  |     | NULL    |                |
| updated_time_stamp | datetime(6)                                              | YES  |     | NULL    |                |
| actual_end_date    | datetime(6)                                              | YES  |     | NULL    |                |
| actual_start_date  | datetime(6)                                              | YES  |     | NULL    |                |
| task_description   | varchar(255)                                             | YES  |     | NULL    |                |
| end_date           | datetime(6)                                              | YES  |     | NULL    |                |
| start_date         | datetime(6)                                              | YES  |     | NULL    |                |
| task_title         | varchar(255)                                             | YES  |     | NULL    |                |
| debug_count        | int                                                      | NO   |     | NULL    |                |
| debug_flag         | bit(1)                                                   | NO   |     | NULL    |                |
| priority           | enum('HIGH','LOW','MEDIUM','VERYHIGH')                   | YES  |     | NULL    |                |
| status             | enum('BACKLOG','DEPLOYMENT','INPROGRESS','TESTING','TODO') | YES  |     | NULL    |                |
| story_point        | int                                                      | NO   |     | NULL    |                |
| testing_flag       | bit(1)                                                   | NO   |     | NULL    |                |
| assigned_by_id     | bigint                                                   | YES  | MUL | NULL    |                |
| assigned_to_id     | bigint                                                   | YES  | MUL | NULL    |                |
| created_by_id      | bigint                                                   | YES  | MUL | NULL    |                |
| project_id         | bigint                                                   | YES  | MUL | NULL    |                |
| sprint_id          | bigint                                                   | YES  | MUL | NULL    |                |
| story_id           | bigint                                                   | YES  | MUL | NULL    |                |
+--------------------+----------------------------------------------------------+------+-----+---------+----------------+
```

Table-7-:Bugs



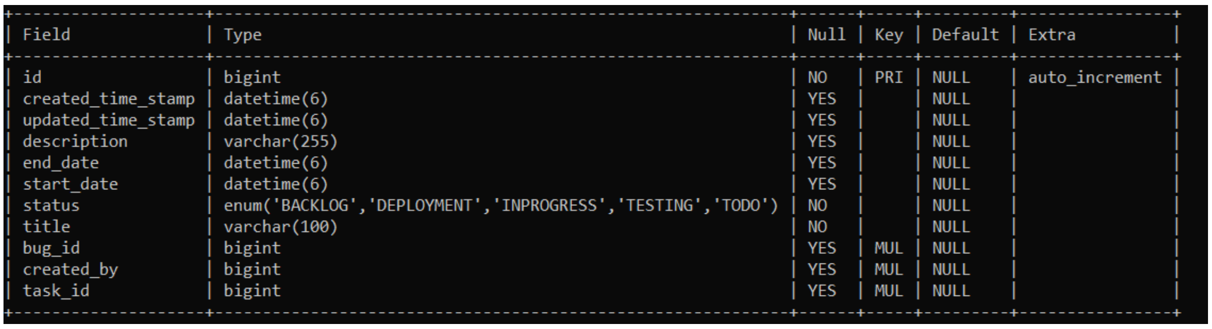
```
+--------------------+-------------------------------------------------------------+------+-----+---------+----------------+
| Field              | Type                                                        | Null | Key | Default | Extra          |
+--------------------+-------------------------------------------------------------+------+-----+---------+----------------+
| id                 | bigint                                                      | NO   | PRI | NULL    | auto_increment |
| created_time_stamp | datetime(6)                                                 | YES  |     | NULL    |                |
| updated_time_stamp | datetime(6)                                                 | YES  |     | NULL    |                |
| actual_end_date    | datetime(6)                                                 | YES  |     | NULL    |                |
| actual_start_date  | datetime(6)                                                 | YES  |     | NULL    |                |
| bug_description    | varchar(255)                                                | YES  |     | NULL    |                |
| end_date           | datetime(6)                                                 | YES  |     | NULL    |                |
| start_date         | datetime(6)                                                 | YES  |     | NULL    |                |
| bug_title          | varchar(255)                                                | YES  |     | NULL    |                |
| priority           | enum('HIGH','LOW','MEDIUM','VERYHIGH')                      | YES  |     | NULL    |                |
| reopen_count       | int                                                         | NO   |     | NULL    |                |
| status             | enum('BACKLOG','DEPLOYMENT','INPROGRESS','TESTING','TODO')  | YES  |     | NULL    |                |
| story_point        | int                                                         | NO   |     | NULL    |                |
| assigned_to        | bigint                                                      | YES  | MUL | NULL    |                |
| project_id         | bigint                                                      | NO   | MUL | NULL    |                |
| reported_by        | bigint                                                      | YES  | MUL | NULL    |                |
| sprint_id          | bigint                                                      | YES  | MUL | NULL    |                |
| story_id           | bigint                                                      | YES  | MUL | NULL    |                |
+--------------------+-------------------------------------------------------------+------+-----+---------+----------------+
```

Table-8-:Sprint_Story (Mapping Table)



```
+-----------+--------+------+-----+---------+-------+
| Field     | Type   | Null | Key | Default | Extra |
+-----------+--------+------+-----+---------+-------+
| story_id  | bigint | NO   | PRI | NULL    |       |
| sprint_id | bigint | NO   | PRI | NULL    |       |
+-----------+--------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

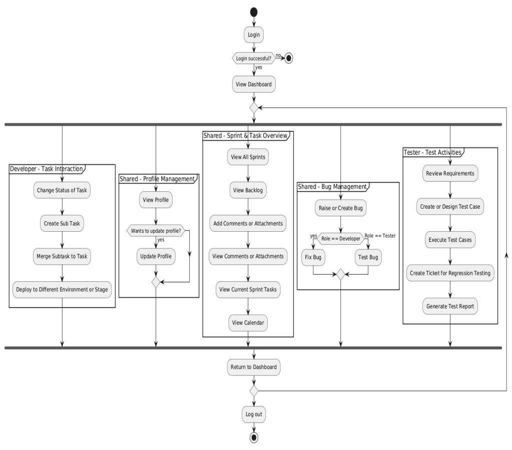
Table-9-:Sub-tasks



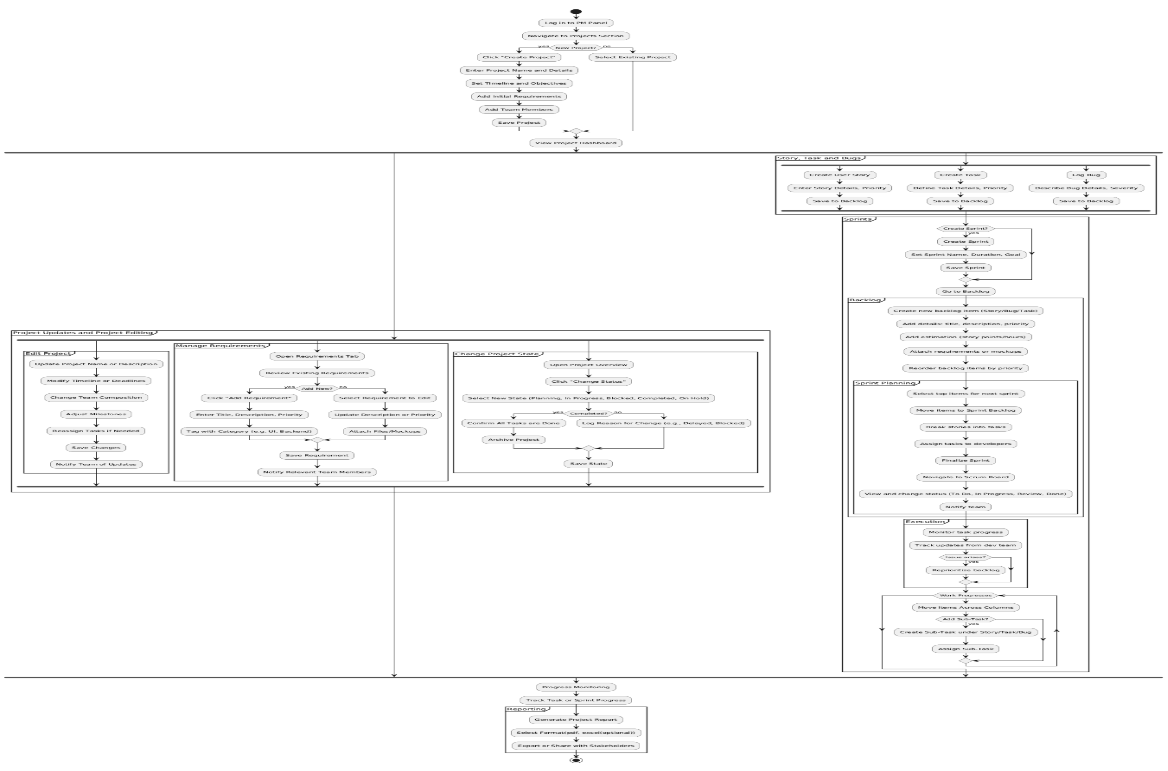
```
+--------------------+-------------------------------------------------------------+------+-----+---------+----------------+
| Field              | Type                                                        | Null | Key | Default | Extra          |
+--------------------+-------------------------------------------------------------+------+-----+---------+----------------+
| id                 | bigint                                                      | NO   | PRI | NULL    | auto_increment |
| created_time_stamp | datetime(6)                                                 | YES  |     | NULL    |                |
| updated_time_stamp | datetime(6)                                                 | YES  |     | NULL    |                |
| description        | varchar(255)                                                | YES  |     | NULL    |                |
| end_date           | datetime(6)                                                 | YES  |     | NULL    |                |
| start_date         | datetime(6)                                                 | YES  |     | NULL    |                |
| status             | enum('BACKLOG','DEPLOYMENT','INPROGRESS','TESTING','TODO')  | NO   |     | NULL    |                |
| title              | varchar(100)                                                | NO   |     | NULL    |                |
| bug_id             | bigint                                                      | YES  | MUL | NULL    |                |
| created_by         | bigint                                                      | YES  | MUL | NULL    |                |
| task_id            | bigint                                                      | YES  | MUL | NULL    |                |
+--------------------+-------------------------------------------------------------+------+-----+---------+----------------+
```

4.2 UML Diagram

Tester & Developer UML-:



Project Manager UML-:


Log in to PM Panel
Navigate to Projects Section
New Project?
yes
no
Click "Create Project"
Select Existing Project
Enter Project Name and Details
Set Timeline and Objectives
Add Initial Requirements
Add Team Members
Save Project
View Project Dashboard
Story, Task and Bugs
Create User Story
Enter Story Details, Priority
Save to Backlog
Create Task
Define Task Details, Priority
Save to Backlog
Log Bug
Describe Bug Details, Severity
Save to Backlog
Sprints
Create Sprint?
yes
Create Sprint
Set Sprint Name, Duration, Goal
Save Sprint
Go to Backlog
Backlog
Create new backlog item (Story/Bug/Task)
Add details: title, description, priority
Add estimation (story points/hours)
Attach requirements or mockups
Reorder backlog items by priority
Sprint Planning
Select top items for next sprint
Move items to Sprint Backlog
Break stories into tasks
Assign tasks to developers
Finalize Sprint
Navigate to Scrum Board
View and change status (To Do, In Progress, Review, Done)
Notify team
Execution
Monitor task progress
Track updates from dev team
Issue arises?
yes
Reprioritize backlog
Work Progresses
Move Items Across Columns
Add Sub-Task?
yes
Create Sub-Task under Story/Task/Bug
Assign Sub-Task
Project Updates and Project Editing
Edit Project
Update Project Name or Description
Modify Timeline or Deadlines
Change Team Composition
Adjust Milestones
Reassign Tasks if Needed
Save Changes
Notify Team of Updates
Manage Requirements
Open Requirements Tab
Review Existing Requirements
Add New?
yes
no
Click "Add Requirement"
Enter Title, Description, Priority
Tag with Category (e.g. UI, Backend)
Select Requirement to Edit
Update Description or Priority
Attach Files/Mockups
Save Requirement
Notify Relevant Team Members
Change Project State
Open Project Overview
Click "Change Status"
Select New State (Planning, In Progress, Blocked, Completed, On Hold)
Completed?
yes
no
Confirm All Tasks are Done
Archive Project
Log Reason for Change (e.g., Delayed, Blocked)
Save State
Progress Monitoring
Track Task or Sprint Progress
Reporting
Generate Project Report
Select Format(pdf, excel(optional))
Export or Share with Stakeholders

---

5. Setup & Installation

This section outlines the requirements and steps to set up Syncora locally for development or deployment.

5.1. Prerequisites

Before installing Syncora, ensure you have the following software installed on your system:

- **Node.js** (v16 or higher) – Download here
- **npm** (comes with Node.js) or **yarn** – Package manager.
- **Java JDK** (v17 or higher) – Required for running Spring Boot backend. Download here
- **Maven** – Build tool for Spring Boot projects. Download here
- **MySQL / PostgreSQL** – Relational database for structured data.
- **MongoDB** – NoSQL database for unstructured data.

- **Git** – For version control. Download here
- **AWS Account** – (Optional for local setup) For S3 bucket and deployment.

5.2. Installation Steps

Step 1: Clone the Repository

git clone https://github.com/CodeBits101/Syncora.git

cd Syncora

Step 2: Install Frontend Dependencies

cd frontend

npm install

or

yarn install

Step 3: Install Backend Dependencies

- **For Spring Boot API**

  cd backend-springboot

  mvn clean install

- **For Node.js Microservices**

  cd backend-node

  npm install

Step 4: Setup Databases

- Create databases in MySQL/PostgreSQL for relational data.
- Start MongoDB service for NoSQL data.

5.3. Configuration

You need to configure environment variables for the application to run properly.

Frontend Configuration (`frontend/.env`)

```
REACT_APP_API_URL=http://localhost:8080

REACT_APP_NODE_SERVICE_URL=http://localhost:5000
```

Spring Boot Backend Configuration (`backend-springboot/src/main/resources/application.properties`)

```
spring.datasource.url=jdbc:mysql://localhost:3306/syncora

spring.datasource.username=YOUR_DB_USERNAME

spring.datasource.password=YOUR_DB_PASSWORD

aws.s3.bucket-name=your-bucket-name

aws.access-key=your-access-key

aws.secret-key=your-secret-key
```

Node.js Backend Configuration (`backend-node/.env`)

```
MONGO_URI=mongodb://localhost:27017/syncora

APP_PORT = 3000
```

- **Start Spring Boot Backend**

  ```
  cd backend-springboot

  mvn spring-boot:run
  ```

- **Start Node.js Microservices**

  ```
  cd backend-node

   npm run dev
  ```

- **Start Frontend**

  ```
  cd frontend

  npm run dev / yarn dev
  ```

---

6. Project Structure

```
Syncora/
│
├── FrontEnd/
│   ├── public/
│   ├── src/
│   │   ├── components/
│   │   ├── pages/
│   │   ├── hooks/
│   │   ├── services/
│   │   ├── utils/
│   │   └── App.js
│   ├── package.json
│   └── .env
│
├── Node-Mongo-Backend/
│   ├── models/
│   ├── routes/
│   ├── controllers/
│   ├── config/
│   ├── utils/
│   ├── server.js
│   ├── package.json
│   └── .env
│
├── Syncora-Spring-Backend/
│   ├── src/main/java/
│   │   └── com/syncora/
│   │       ├── controllers/        # REST API controllers
│   │       ├── services/           # Business logic layer
│   │       ├── repositories/        # Database access (JPA/Hibernate)
│   │       ├── models/             # Entity classes
│   │       ├── dtos/             # Data Transfer Objects
│   │       ├── securities/         # Security config (JWT, Auth, etc.)
│   │       └── config/            # App and AWS configurations
│   ├── src/main/resources/
│   ├── pom.xml
│   └── target/
│
├── Project-Utils/
│
├── db/
│
├── README.md
├── info.txt
├── package.json
└── yarn.lock
```

---

7. Usage Guide

7.1 Running the Application

Frontend (React.js)

Development Mode

```
cd FrontEnd
yarn install         # Install dependencies
yarn start           # Start development server
```

The frontend will run at http://localhost:5173.

Production Build

```
cd FrontEnd
yarn install
yarn build           # Builds the app for production
serve -s build       # Serve the production build locally
```

Node-Mongo Backend (Microservices)

Development Mode

```
cd Node-Mongo-Backend
npm install          # Install dependencies
npm run dev          # Start in development mode
```

Runs by default at http://localhost:3000 (check `.env` for exact port).

Production Mode

cd Node-Mongo-Backend

npm install

npm start

Spring Boot Backend (Java)

Development Mode

cd Syncora-Spring-Backend

mvn spring-boot:run

Runs by default at http://localhost:8080 (configured in `application.properties`).

Production Mode

cd Syncora-Spring-Backend

mvn clean package

java -jar target/Syncora-0.0.1-SNAPSHOT.jar

7.2 API Endpoints

Below are some sample API endpoints provided by Syncora backends.

Spring Boot Backend APIs

Method	Endpoint	Description	Request Body Example	Response Example
`POST`	`/employees/login`	User authentication (JWT token)	`{` `"email": "string",` `"password": "string"`	`{ "token": "jwt_token_here" }`

			`}`	
`POST`	`/projects`	Create a new project	`{ "title": "string", "description": "string", …..}`	`{ "id": 1, "name": "New Project", ... }`
`GET`	`/projects/{id}`	Get project details by ID	—	`{ "id": 1, "name": "New Project", ... }`
`PUT`	`/projects/{id}`	Update project details	`{ "name": "Updated Name".... }`	`{ "id": 1, "name": "Updated Name", ... }`
`DELETE`	`/projects/{id}`	Delete a project	—	`{ "message": "Project deleted successfully" }`

Node-Mongo Backend APIs

Method	Endpoint	Description	Request Body Example	Response Example

`GET`	{{base_url}}/dev/	Get All dev data based on the lang key	—	`{ "data": "whole data like name dob and handles" }`
`GET`	{{base_url}}/about	Retrieve Project Data	—	`{ "data": "data like project features" }`
`GET`	{{base_url}}/lang	Get lang options	—	`{ "message": "File deleted" }`

⚡ **Tip:**

- Make sure `.env` files are properly configured before running the application (DB URIs, AWS credentials, JWT secrets, ports).
- If running both backends together, confirm there are no **port conflicts**.

---

8. Best Practices

8.1 Coding Standards

- **Naming Conventions**
  - **Java (Spring Boot)**:
    - Classes: `PascalCase` (e.g., `ProjectService`)
    - Variables & Methods: `camelCase` (e.g., `projectName`, `getProjectDetails()`)
    - Constants: `UPPER_CASE_WITH_UNDERSCORES` (e.g., `MAX_TASKS_LIMIT`)

    - **JavaScript/React (Frontend & Node)**:
        - Components: `PascalCase` (e.g., `TaskList`)
        - Functions & Variables: `camelCase` (e.g., `fetchProjects`)
        - File names: Match component name (e.g., `TaskList.js`)
- **Code Documentation**
    - Use **Javadoc** in Spring Boot for service, controller, and DTO classes.
    - Use **JSDoc** for Node.js services and utility functions.
    - Maintain clear inline comments for complex logic.
- **Folder Structure Discipline**
    - Keep **DTOs, Controllers, Services, and Repositories** separated in Spring Boot.
    - In React, maintain `components`, `pages`, `hooks`, and `utils` separation.

8.2 Security Measures

- **Authentication & Authorization**
    - Use **JWT tokens** for secure API access.
    - Role-based access control (Admin, Developer, Tester) enforced at backend.
- **Input Validation**
    - Validate all incoming request bodies using:
- **Data Protection**
    - Store sensitive configuration in `.env` files (never commit to Git).
    - Encrypt passwords using **BCrypt** before storage.
    - Use HTTPS in production.

8.3 Performance Optimization

- **Backend**
  - Use pagination for large data sets in project/task listing APIs.
  - Cache frequently accessed data (e.g., using Spring Cache or Redis).
  - Optimize database queries — use **indexes** for frequently queried fields.
- **Frontend**
  - Implement **lazy loading** for components and routes.
  - Use **React.memo** and **useCallback** to prevent unnecessary re-renders.
- **Database**
  - Separate relational (SQL) and unstructured (MongoDB) workloads appropriately.
- **File Storage**
  - Store only references/URLs in the database, not raw file binaries.

---

9. Troubleshooting & FAQs

9.1.Trouble Shooting Table -:

Issue	Possible Cause	Solution
Frontend not starting (`yarn start` fails)	Missing dependencies or incorrect Node.js version	Ensure Node.js ≥ 16.x is installed. Run `yarn install` again. Delete `node_modules` and retry if needed.
Spring Boot backend fails to start	Database connection failure	Check `application.properties` for correct DB credentials. Ensure SQL and MongoDB servers are running.

Node-Mongo backend not serving About/FAQs data	API route not reachable or service down	Verify Node backend is running on the configured port. Check `.env` for `API_BASE_URL`.
Customer Support data not appearing	Missing API endpoint or frontend call error	Confirm `/api/support` route is working in Node backend. Check browser network tab for errors.
FAQs not updating on frontend	Backend data not refreshed or endpoint mismatch	Make sure `/api/faqs` returns latest data and frontend is pointing to correct backend URL.
CORS errors when calling Node APIs	Cross-origin restrictions in backend	Add frontend origin to CORS whitelist in Node.js backend configuration.

9.2 Common FAQs-:

Q1: What is Syncora used for?
A: Syncora is a project and task management platform designed to manage the full Software Development Life Cycle (SDLC) — from requirements gathering to deployment — enabling seamless collaboration between teams.

Q2: Who can use Syncora?
A: Syncora supports three main user roles:
-Admin → Manages users, projects, and overall settings.
-Developer → Works on assigned tasks and updates progress.
-Tester → Validates tasks, reports bugs, and ensures quality.

---

Q3: How does Syncora store and manage project files?
A: All files and documents are stored securely in “AWS S3 buckets”, ensuring reliability, scalability, and high availability.

---

Q4: How will I be notified about changes in my projects or tasks?
A: Syncora sends automated email notifications for task assignments, status changes, and approaching deadlines through its built-in mail service.

Q5: Does Syncora provide tools to track project progress visually?
A: Yes. Syncora integrates advanced visualization tools like Gantt charts, calendars, and dashboards using SyncFusion, allowing you to monitor timelines, milestones, and overall progress.

---

10. References & Resources

Below are key references and resources used in building and maintaining Syncora:

Official Documentation

- **React.js** → https://react.dev/
- **Material UI** → https://mui.com/
- **Framer Motion** → https://www.framer.com/motion/
- **SyncFusion Components** → https://www.syncfusion.com/
- **Spring Boot** → https://spring.io/projects/spring-boot
- **Node.js** → https://nodejs.org/en/docs/

Additional Frontend Libraries & UI Resources

- **Formik** →
- **Tailwind CSS** → https://tailwindcss.com/
- **React Bootstrap** → https://react-bootstrap.netlify.app/

Databases & Storage

- **MongoDB** → https://www.mongodb.com/docs/

- **SQL (MySQL)** → https://dev.mysql.com/doc/
- **AWS S3 Bucket** → https://docs.aws.amazon.com/s3/
- **Cloudinary (Media Storage)** → https://cloudinary.com/

Cloud & Infrastructure

- **AWS EC2** → https://docs.aws.amazon.com/ec2/
- **Clean Architecture Principles** → https://blog.cleancoder.com/
- **SOLID Design Principles** → https://en.wikipedia.org/wiki/SOLID

Development Tools & Community

- **Git** → https://git-scm.com/doc
- **Postman (API Testing)** → https://learning.postman.com/
- **VS Code** → https://code.visualstudio.com/docs
- **Stack Overflow** → https://stackoverflow.com/questions

---

11. Contact & Support

If you have any questions, encounter issues, or wish to request new features, feel free to reach out through the following support channels:

📩 **Email Support**
General Inquiries & Feedback: C. Bits

Technical Assistance: C. Bits

📞 **Phone Support**
General Support: +91 9301667800 , +91 7972751012 ,+91 8788449649 ,+91 7057645314

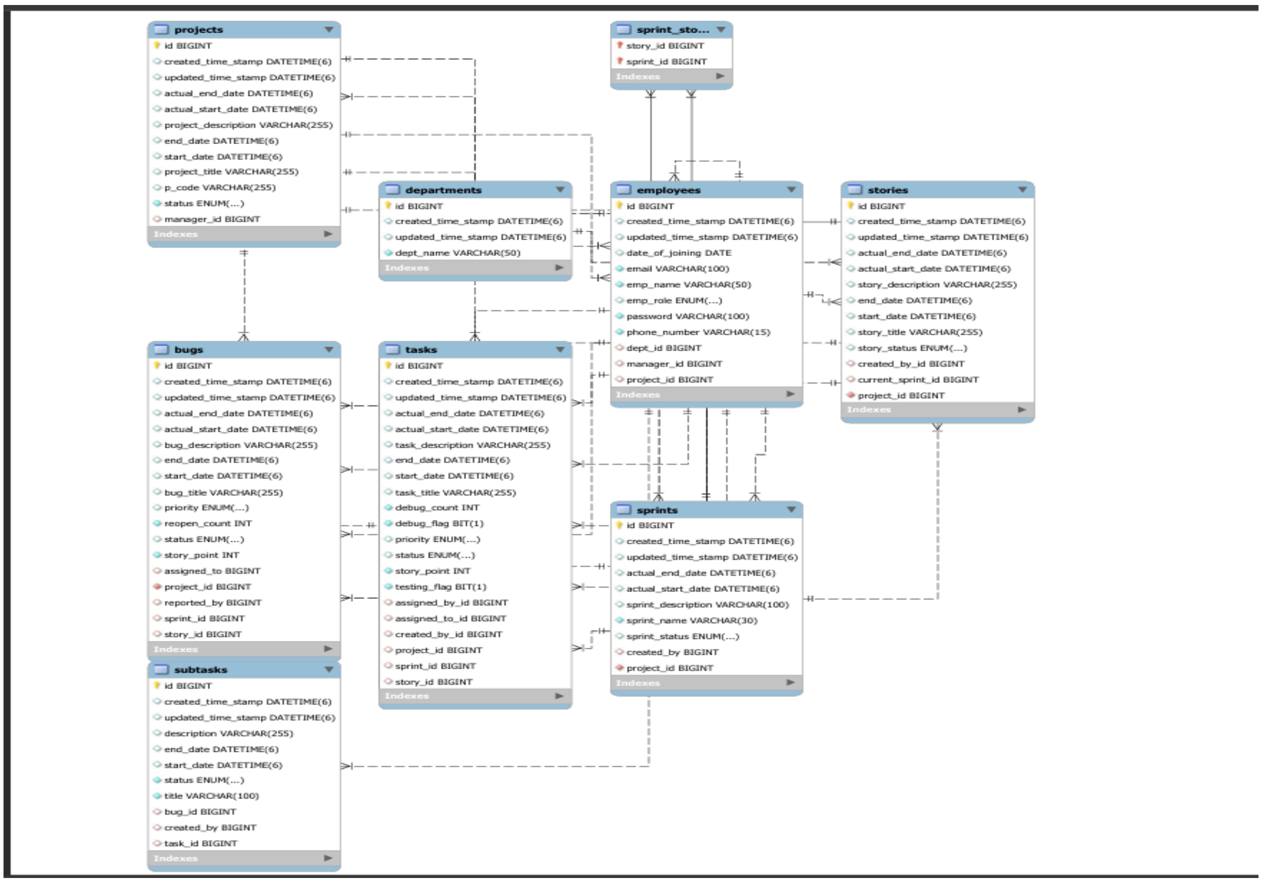
Technical Assistance: +91 9301667800 , +91 7972751012 ,+91 8788449649 ,+91 7057645314

⏱ Support Hours

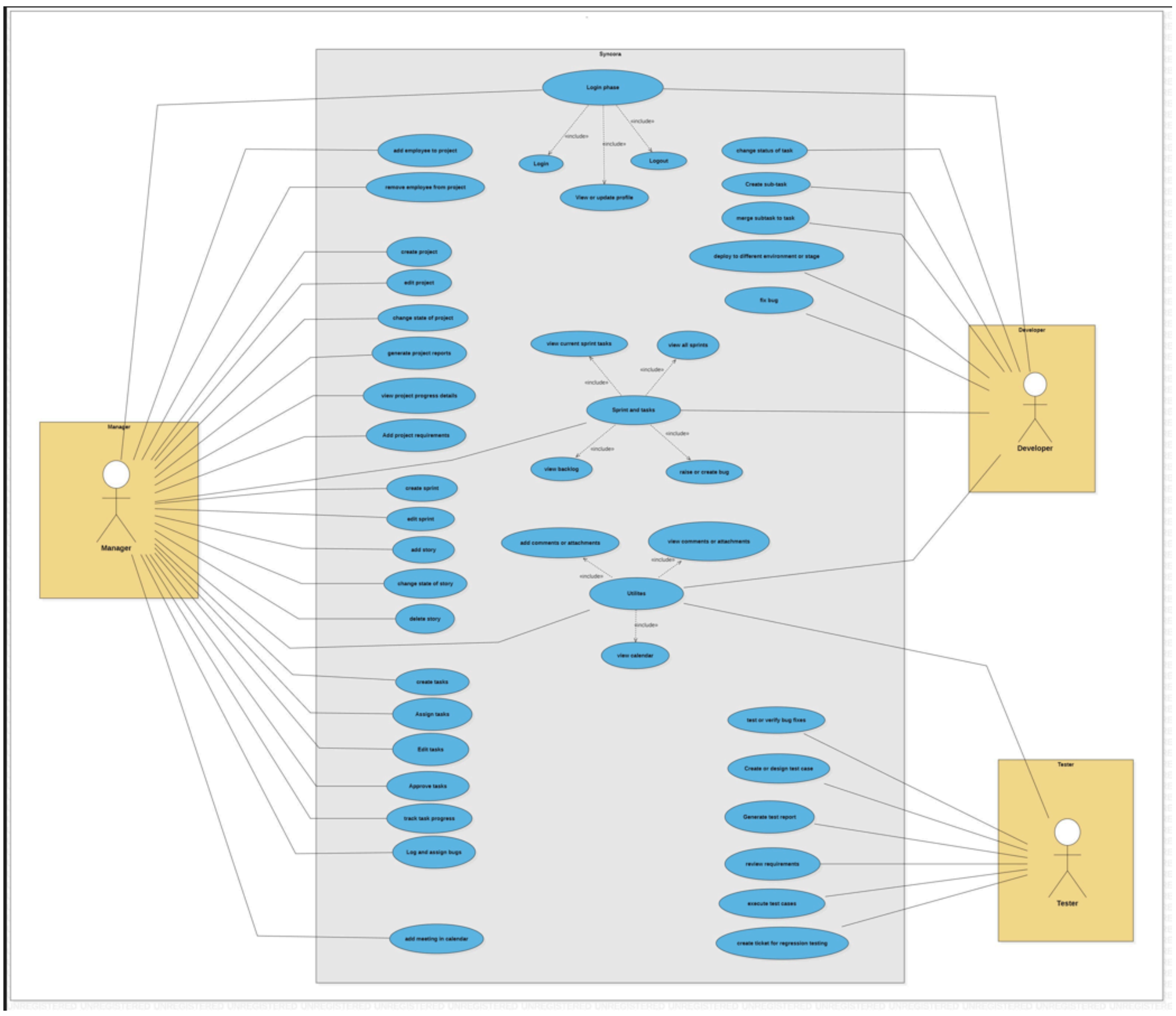
- **Monday – Friday**: 9:00 AM – 6:00 PM (GMT)
- **Response Time**: Within 24–48 hours on business days

12.Appendix A

Class Diagram

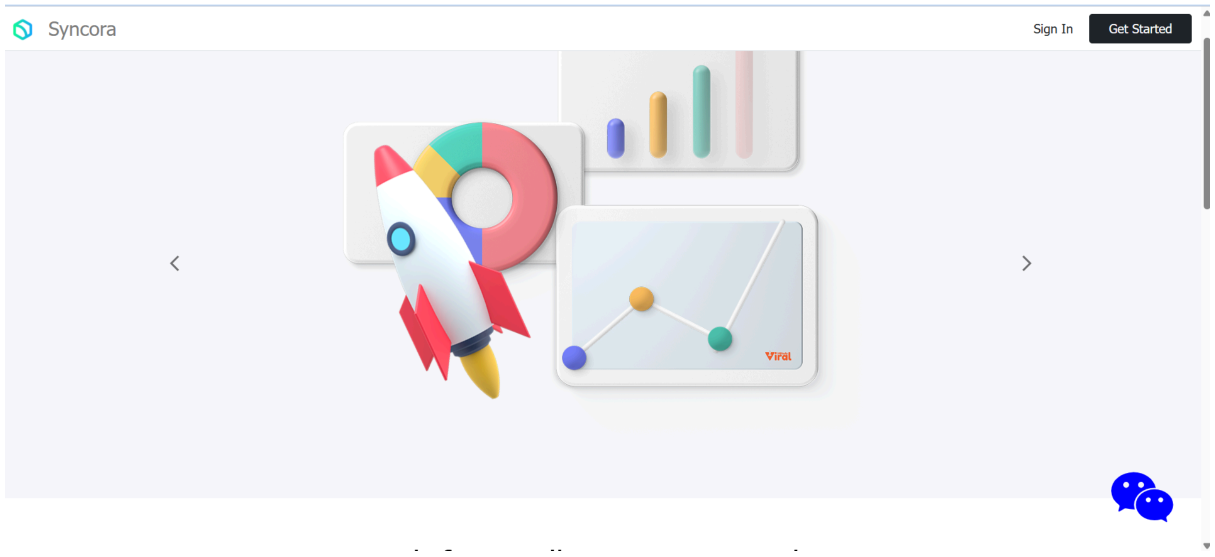


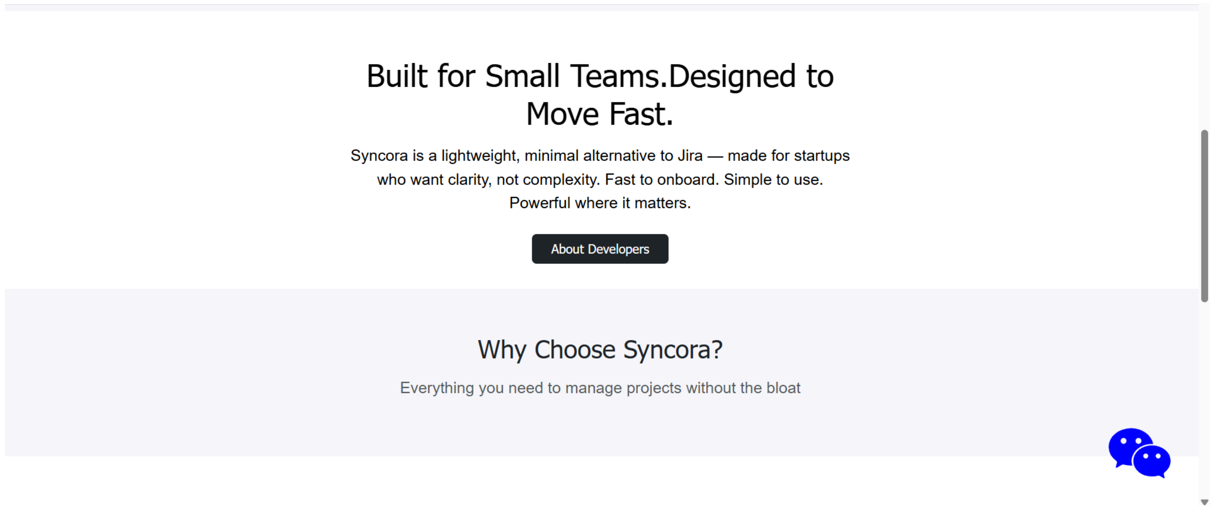
<u>Data-Flow Diagram</u>



13.Appendix B

Home Page-:





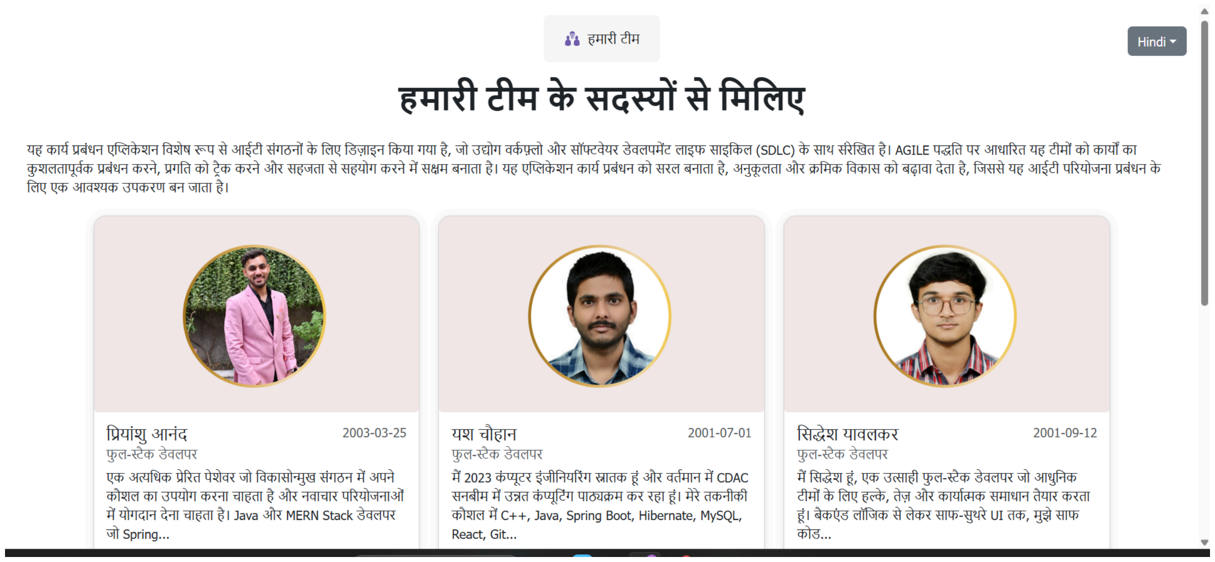
Built for Small Teams.Designed to Move Fast.

Syncora is a lightweight, minimal alternative to Jira — made for startups who want clarity, not complexity. Fast to onboard. Simple to use. Powerful where it matters.

Why Choose Syncora?

Everything you need to manage projects without the bloat

2.Globalisation (About Developers Page) -:



हमारी टीम के सदस्यों से मिलिए

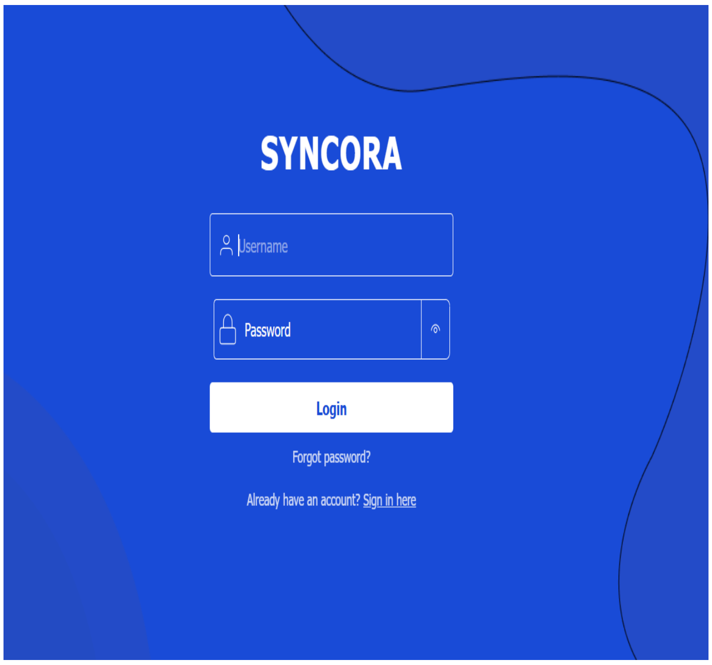
यह कार्य प्रबंधन एप्लिकेशन विशेष रूप से आईटी संगठनों के लिए डिज़ाइन किया गया है, जो उद्योग वर्कफ़्लो और सॉफ्टवेयर डेवलपमेंट लाइफ साइकिल (SDLC) के साथ संरेखित है। AGILE पद्धति पर आधारित यह टीमों को कार्यों का कुशलतापूर्वक प्रबंधन करने, प्रगति को ट्रैक करने और सहजता से सहयोग करने में सक्षम बनाता है। यह एप्लिकेशन कार्य प्रबंधन को सरल बनाता है, अनुकूलता और क्रमिक विकास को बढ़ावा देता है, जिससे यह आईटी परियोजना प्रबंधन के लिए एक आवश्यक उपकरण बन जाता है।

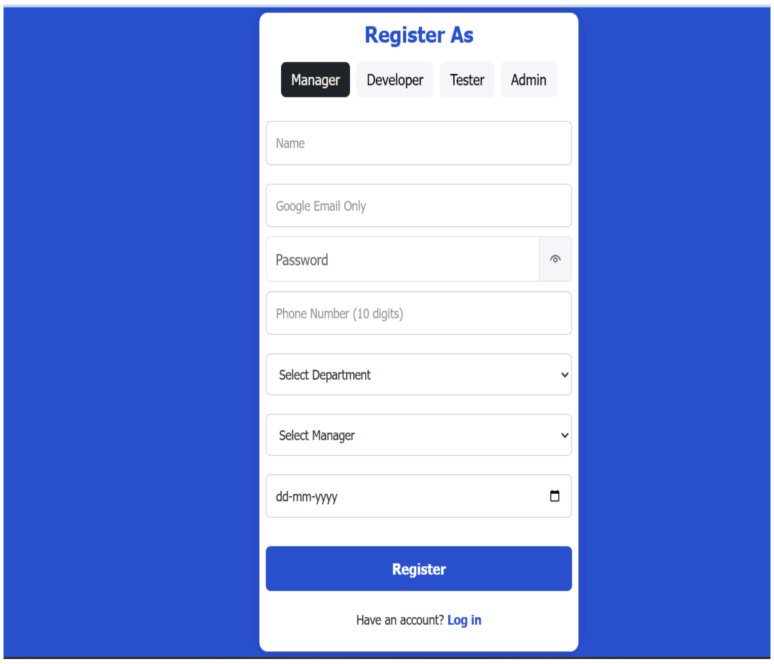
प्रियांशु आनंद 2003-03-25
फुल-स्टैक डेवलपर
एक अत्यधिक प्रेरित पेशेवर जो विकासोन्मुख संगठन में अपने कौशल का उपयोग करना चाहता है और नवाचार परियोजनाओं में योगदान देना चाहता है। Java और MERN Stack डेवलपर जो Spring...

यश चौहान 2001-07-01
फुल-स्टैक डेवलपर
मैं 2023 कंप्यूटर इंजीनियरिंग स्नातक हूं और वर्तमान में CDAC सनबीम में उन्नत कंप्यूटिंग पाठ्यक्रम कर रहा हूं। मेरे तकनीकी कौशल में C++, Java, Spring Boot, Hibernate, MySQL, React, Git...

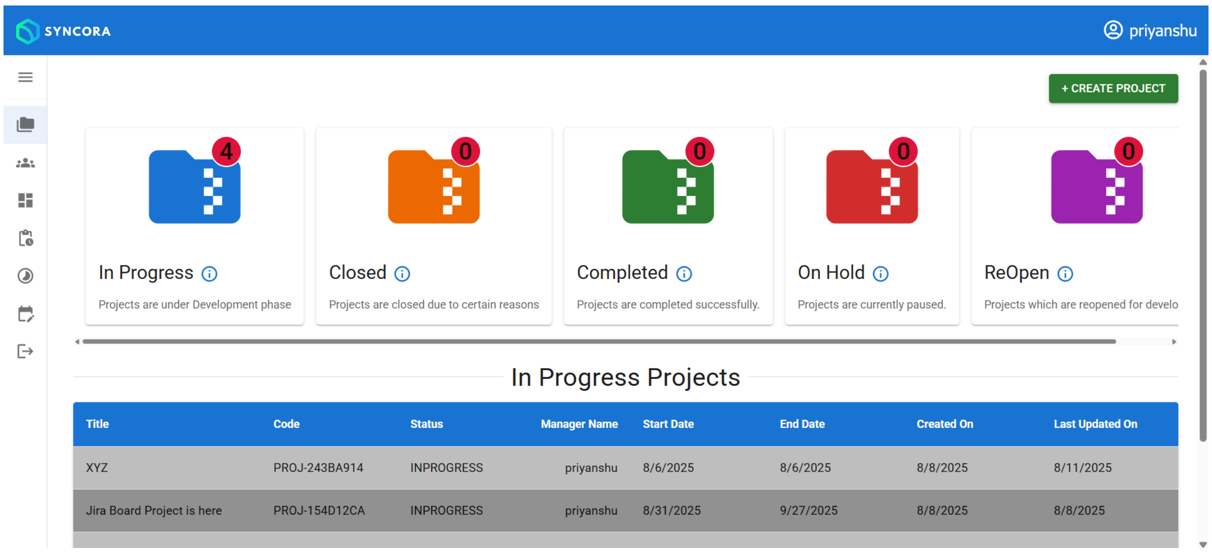
सिद्धेश यावलकर 2001-09-12
फुल-स्टैक डेवलपर
मैं सिद्धेश हूं, एक उत्साही फुल-स्टैक डेवलपर जो आधुनिक टीमों के लिए हल्के, तेज़ और कार्यात्मक समाधान तैयार करता हूं। बैकएंड लॉजिक से लेकर साफ-सुथरे UI तक, मुझे साफ कोड...

3.Register and Login Page





4.Manager Home Screen -:



In Progress
Projects are under Development phase

Closed
Projects are closed due to certain reasons

Completed
Projects are completed successfully.

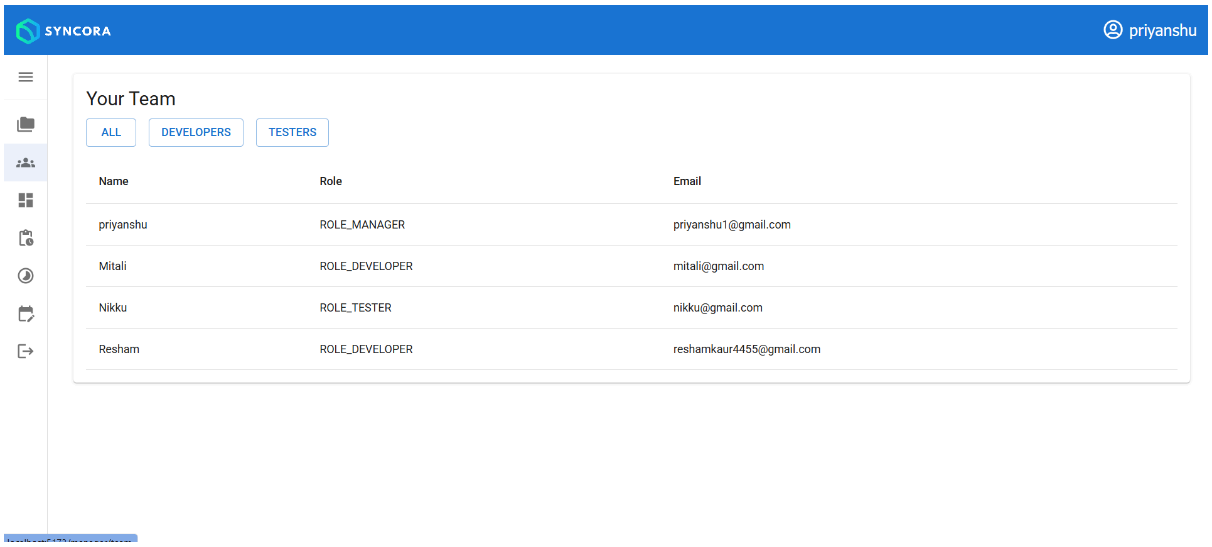
On Hold
Projects are currently paused.

ReOpen
Projects which are reopened for develo

In Progress Projects

Title	Code	Status	Manager Name	Start Date	End Date	Created On	Last Updated On
XYZ	PROJ-243BA914	INPROGRESS	priyanshu	8/6/2025	8/6/2025	8/8/2025	8/11/2025
Jira Board Project is here	PROJ-154D12CA	INPROGRESS	priyanshu	8/31/2025	9/27/2025	8/8/2025	8/8/2025

5.Manager Team Page -:

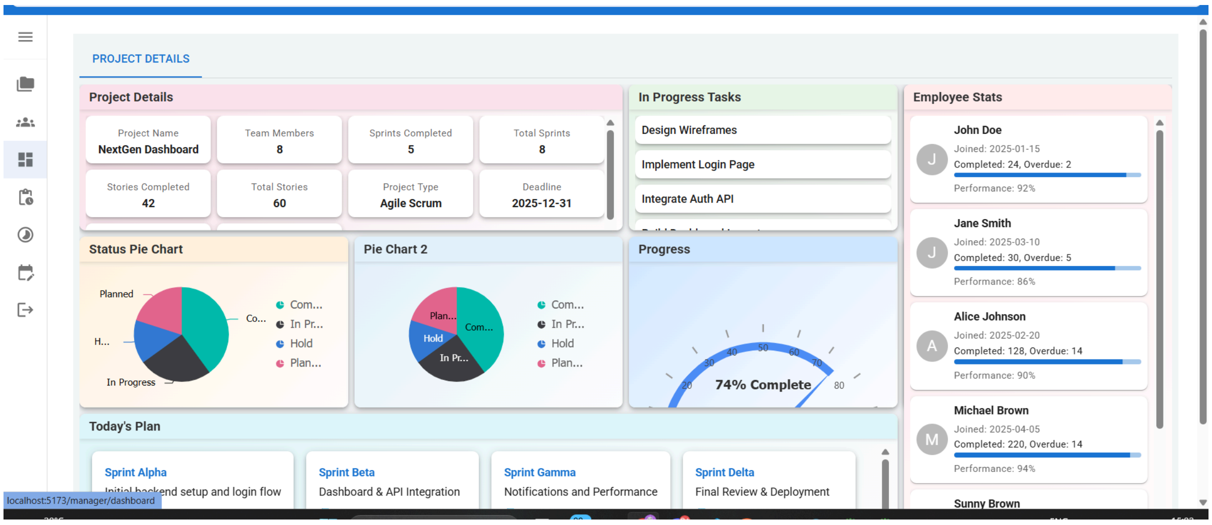


Your Team

ALL DEVELOPERS TESTERS

Name	Role	Email
priyanshu	ROLE_MANAGER	priyanshu1@gmail.com
Mitali	ROLE_DEVELOPER	mitali@gmail.com
Nikku	ROLE_TESTER	nikku@gmail.com
Resham	ROLE_DEVELOPER	reshamkaur4455@gmail.com

6.DashBoard for particular Project -:



Project Details

Project Name NextGen Dashboard	Team Members 8	Sprints Completed 5	Total Sprints 8
Stories Completed 42	Total Stories 60	Project Type Agile Scrum	Deadline 2025-12-31

In Progress Tasks

- Design Wireframes
- Implement Login Page
- Integrate Auth API

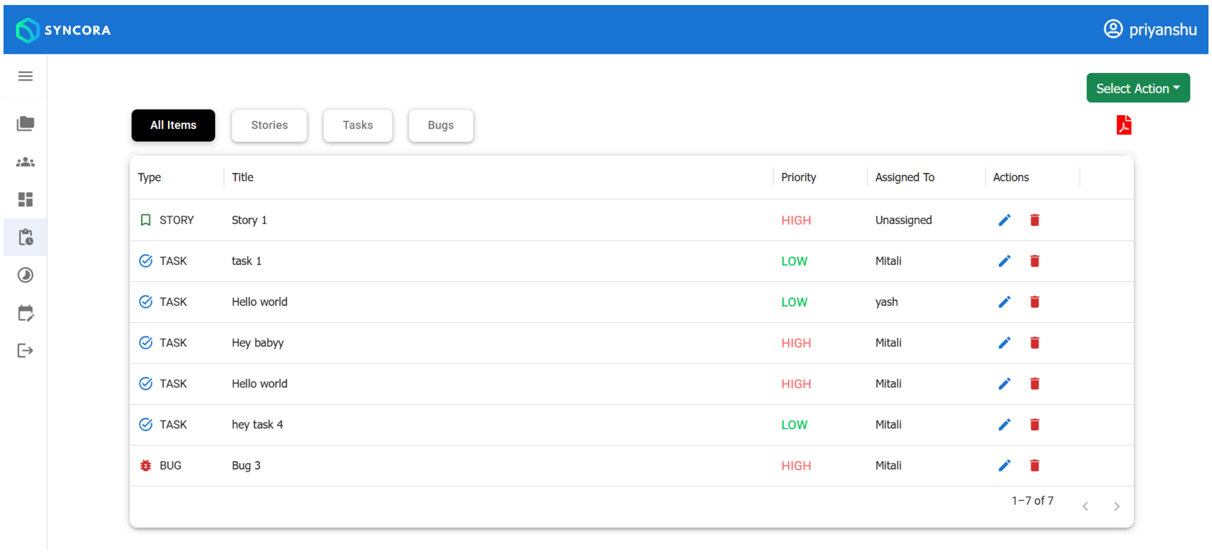
Employee Stats

Name	Joined	Completed	Overdue	Performance
John Doe	2025-01-15	24	2	92%
Jane Smith	2025-03-10	30	5	86%
Alice Johnson	2025-02-20	128	14	90%
Michael Brown	2025-04-05	220	14	94%
Sunny Brown				

Today's Plan

- Sprint Alpha — Initial backend setup and login flow
- Sprint Beta — Dashboard & API Integration
- Sprint Gamma — Notifications and Performance
- Sprint Delta — Final Review & Deployment

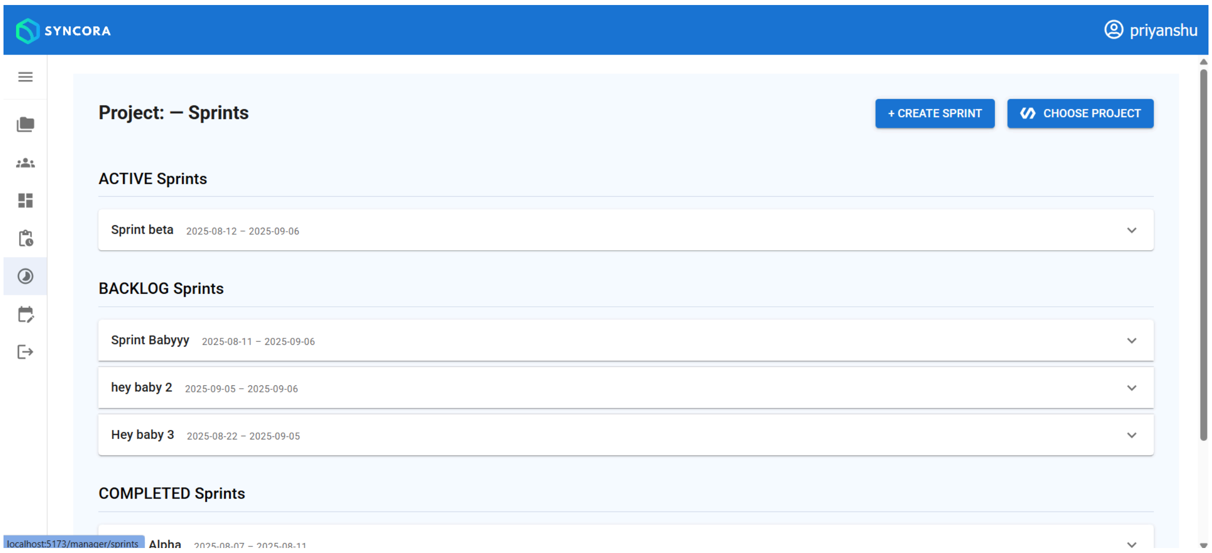
7.BackLog Page -:



Type	Title	Priority	Assigned To	Actions
STORY	Story 1	HIGH	Unassigned	
TASK	task 1	LOW	Mitali	
TASK	Hello world	LOW	yash	
TASK	Hey babyy	HIGH	Mitali	
TASK	Hello world	HIGH	Mitali	
TASK	hey task 4	LOW	Mitali	
BUG	Bug 3	HIGH	Mitali	

1–7 of 7

8.Sprints Page-:



Project: — Sprints

ACTIVE Sprints

Sprint beta 2025-08-12 – 2025-09-06

BACKLOG Sprints

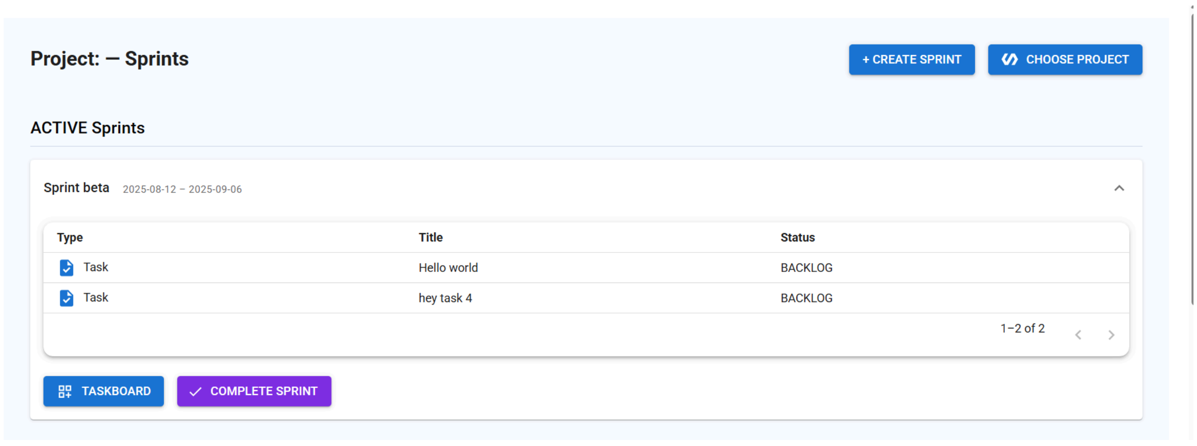
Sprint Babyyy 2025-08-11 – 2025-09-06

hey baby 2 2025-09-05 – 2025-09-06

Hey baby 3 2025-08-22 – 2025-09-05

COMPLETED Sprints

Alpha 2025-08-07 – 2025-08-11



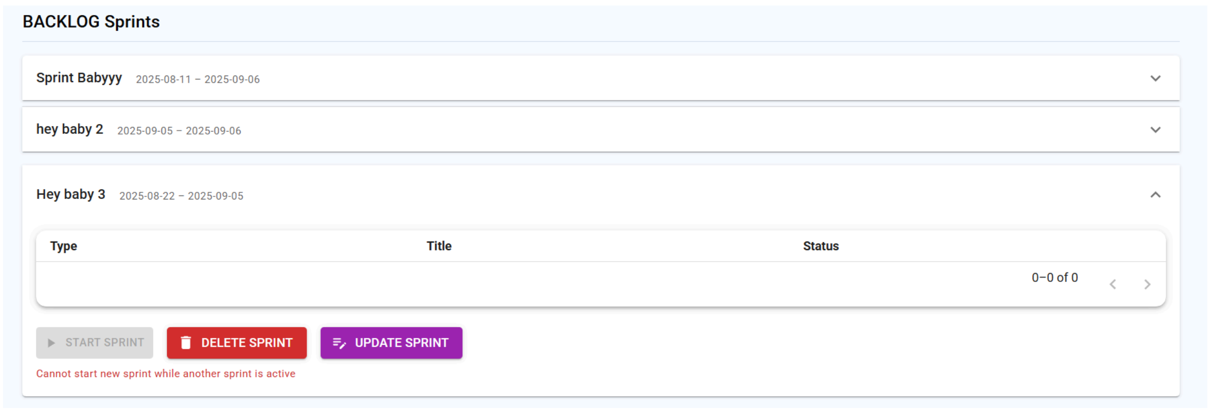
Project: — Sprints

ACTIVE Sprints

Sprint beta 2025-08-12 – 2025-09-06

Type	Title	Status
Task	Hello world	BACKLOG
Task	hey task 4	BACKLOG

1–2 of 2



BACKLOG Sprints

Sprint Babyyy 2025-08-11 – 2025-09-06

hey baby 2 2025-09-05 – 2025-09-06

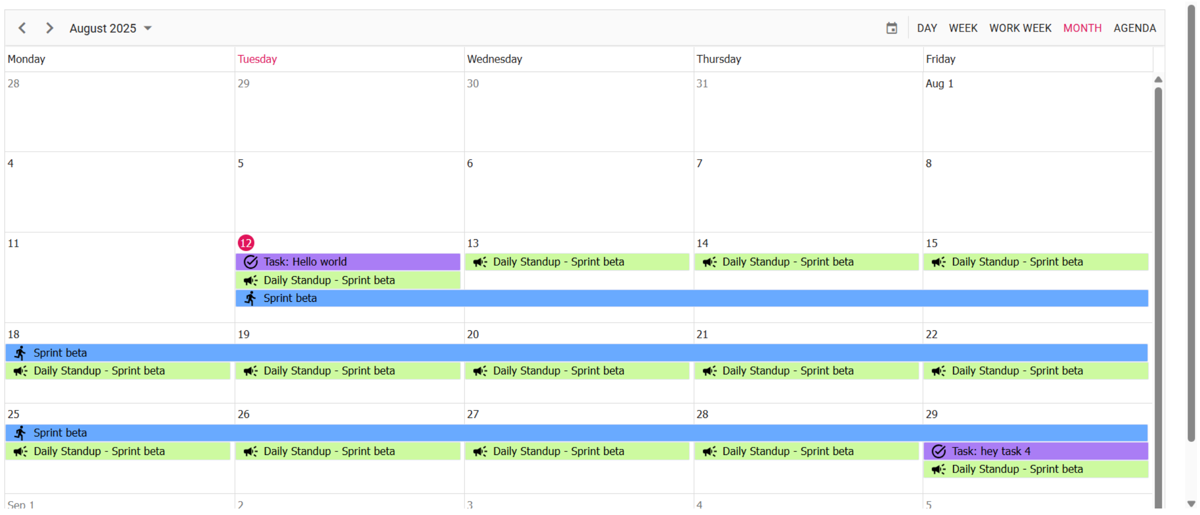
Hey baby 3 2025-08-22 – 2025-09-05

Type	Title	Status

0–0 of 0

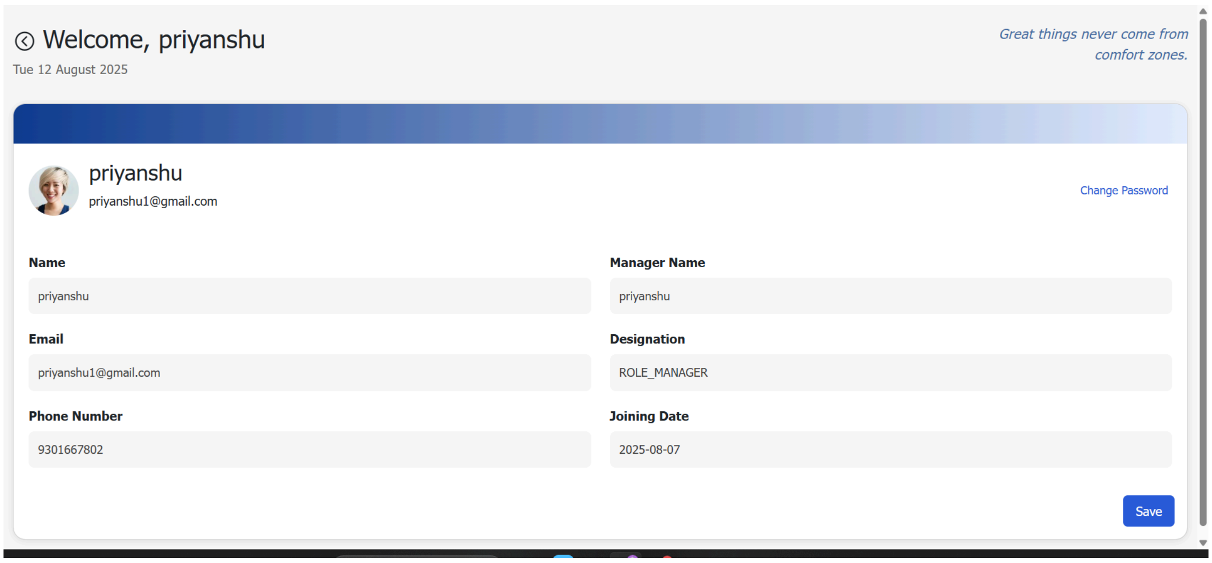
Cannot start new sprint while another sprint is active

9.Calendar Page-:



Monday	Tuesday	Wednesday	Thursday	Friday
28	29	30	31	Aug 1
4	5	6	7	8
11	12 Task: Hello world Daily Standup - Sprint beta Sprint beta	13 Daily Standup - Sprint beta	14 Daily Standup - Sprint beta	15 Daily Standup - Sprint beta
18 Sprint beta Daily Standup - Sprint beta	19 Daily Standup - Sprint beta	20 Daily Standup - Sprint beta	21 Daily Standup - Sprint beta	22 Daily Standup - Sprint beta
25 Sprint beta Daily Standup - Sprint beta	26 Daily Standup - Sprint beta	27 Daily Standup - Sprint beta	28 Daily Standup - Sprint beta	29 Task: hey task 4 Daily Standup - Sprint beta
Sep 1	2	3	4	5

10.Profile Page-:



Welcome, priyanshu

Tue 12 August 2025

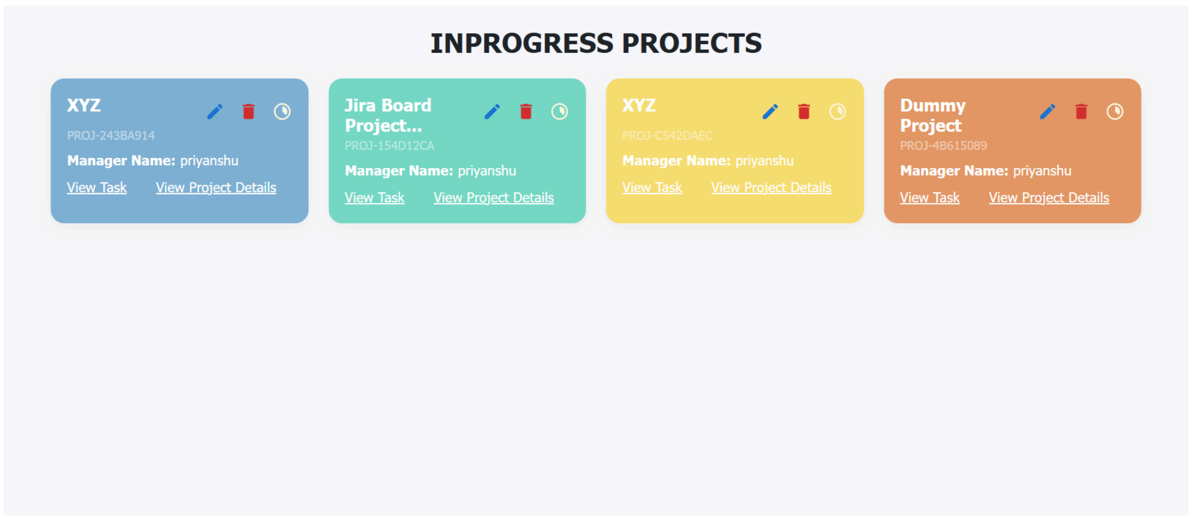
Great things never come from comfort zones.

priyanshu

priyanshu1@gmail.com

Name	priyanshu
Manager Name	priyanshu
Email	priyanshu1@gmail.com
Designation	ROLE_MANAGER
Phone Number	9301667802
Joining Date	2025-08-07

11.Project-Status Wise Details -:



INPROGRESS PROJECTS

XYZ
PROJ-243BA914
Manager Name: priyanshu
View Task View Project Details

Jira Board Project...
PROJ-154D12CA
Manager Name: priyanshu
View Task View Project Details

XYZ
PROJ-C542DAEC
Manager Name: priyanshu
View Task View Project Details

Dummy Project
PROJ-4B615089
Manager Name: priyanshu
View Task View Project Details