

Sri Lanka Institute of Information Technology



Project Proposal Report
Information Technology Project (IT2080)
2025
ITP_2025_Y2_S2_NU10

ProJectra – Project Management App

Submitted by:

	Name with Initials	Registration Number	Contact Phone Number	Email
1.	K.Seyon	IT2375272	07139931	it23752672@my.sliit.lk
2.	K.Braveenraj	IT23204416	0766611829	it23204416@my.sliit.lk
3.	N.Vinushan	IT22545862	077 5263630	it22545862@my.sliit.lk
4.	P.Sinthujan	IT23856240	0753621713	it23856240@my.sliit.lk
5.	S.Rishikeshan	IT23778344	074 0110451	it23778344@my.sliit.lk

Table of Contents

Background	4
Nature of the Business.....	4
Startup Vision.....	4
Target Users	5
Problems and motivations	6
Problems.....	6
Motivation	6
Aims and Objectives	7
Aims.....	7
Objectives.....	8
System Overview	9
System Introduction	9
Functional Requirements	10
1. Member management.....	10
2. Project management.....	11
3. User Management	12
4. Task Management	12
5. App management	13
Nonfunctional Requirements	14
Technical Requirements.....	15
Literature Review	17
Methodology	20
Requirements Engineering Methods	20
Design Methods	20
Tools and Technologies.....	21
Integration Methods.....	22
Agile Software Engineering Methodology	23
Motives behind the adoption of Agile Methodology.....	24
Conclusion:	26
Project plan (Gantt chart)	27
Work breakdown structure (Work distribution)	28

Evaluation Method.....	30
Evaluation Criteria	30
Testing Methods.....	31
Reference	32
Appendix.....	33

Background

Nature of the Business

In today's fast-paced, interconnected world, effective project management is critical for organizations of all sizes. Whether it's a small startup coordinating its first product launch or a large enterprise managing multiple teams across time zones, the ability to plan, track, and deliver work efficiently can be the difference between success and failure.

However, many existing project management tools face a common set of challenges:

- They are too complex for small or non-technical teams, leading to poor adoption.
- They lack industry-specific adaptability, forcing teams to work around rigid workflows.
- They fail to centralize core operations, requiring organizations to juggle multiple disconnected tools for members, tasks, users, and administration.

Projectra is a startup-driven SaaS (Software as a Service) platform designed to bridge these gaps. Our vision is to create a humanized, intuitive, and scalable project management ecosystem that adapts to teams rather than forcing teams to adapt to the tool.

By integrating Member Management, Project Management, User Management, Task Management, and App Management into one platform, Projectra will streamline operations, improve collaboration, and enhance decision-making. Our solution will be accessible on desktop, tablet, and mobile with a consistent, responsive interface for seamless usage anywhere.

Startup Vision

Projectra is not just another project management tool; it's a collaboration hub built for the modern hybrid workspace. We aim to simplify team operations without sacrificing the power and flexibility required by large-scale projects.

Our vision includes:

- Unifying project operations under a single platform — no more switching between apps for tasks, members, and admin controls.
- Humanizing digital workspaces — making tools feel personal, approachable, and motivating rather than rigid and corporate.
- Adapting to industry needs — customizable workflows that suit technology, creative, marketing, construction, education, and other industries.
- Scalable architecture — growing alongside teams from 3 members to 3,000 without degrading performance.
- Data-driven decision-making — integrated analytics to measure productivity, project timelines, and resource usage.

Target Users

The target audience of the Project management app are project manager or SE , Vendors.

1. Team Members

These are the individuals responsible for executing specific tasks within projects.

- Key Needs: Clear task assignments, real-time updates, collaboration tools, and visibility into project timelines.
- Example Use Case: A marketing associate working on a campaign can view their assigned tasks, communicate with teammates via comments, upload required files, and track deadlines in real time.

2. Project Managers / Team Leads

These users are responsible for overseeing multiple projects, assigning resources, and ensuring deadlines are met.

- Key Needs: Centralized project dashboards, milestone tracking, workload balancing, and the ability to generate progress reports.
- Example Use Case: A software development lead manages three concurrent product sprints, assigns user stories to developers, monitors task completion rates, and generates weekly reports for stakeholders.

3. Organization Administrators

These users have high-level control over the platform and its configuration.

- Key Needs: Secure user authentication, role-based access control, integration management, subscription oversight, and organization-wide analytics.
- Example Use Case: An IT admin in a mid-sized company configures company branding within Projectra, manages access permissions for 120 employees, and integrates the platform with Google Workspace for single sign-on.

Problems and motivations

Problems

Our preliminary market research revealed several recurring pain points experienced by organizations when using existing project management tools:

1. **Fragmented Workflows**
 - Many organizations juggle multiple tools for different purposes — for example, using one tool for task tracking, another for communication, and yet another for reporting.
 - This fragmentation leads to inefficiency, duplicated work, and poor visibility across teams.
2. **Overly Complex or Overly Simplistic Solutions**
 - High-end enterprise tools like Jira can overwhelm smaller teams due to steep learning curves.
 - Conversely, overly simple tools like Trello lack the advanced reporting and customization features required for complex projects.
3. **Limited Adaptability**
 - Many tools force teams into rigid workflows that don't align with their industry or operational style.
 - Customization is often possible but requires costly development or premium plans.
4. **Inefficient Communication**
 - Delays occur when task updates or changes in project status are not communicated instantly to relevant stakeholders.
 - Lack of built-in collaboration features forces teams to rely on external chat tools, further fragmenting communication.
5. **Weak Administration Controls**
 - Organization-wide settings, permission management, and compliance tracking are often lacking or hidden behind paywalls.
 - This creates security vulnerabilities and limits the ability of large organizations to maintain control.

Motivation

The **global shift to remote and hybrid work** has highlighted the urgent need for centralized, adaptive, and user-friendly project management solutions. Research by Gartner (2024) indicates that over **60% of organizations** struggle with tool adoption due to poor usability and lack of integration between systems.

Projectra aims to address this by:

- **Unifying Core Functions**
Integrating Member, Project, Task, User, and App Management in one platform eliminates the need for multiple disconnected tools.
- **Balancing Simplicity and Power**
Designing an interface that is approachable for beginners yet equipped with advanced features for complex project needs.
- **Customizable Workflows**
Allowing organizations to define their own processes, approval steps, and reporting formats without requiring costly custom development.
- **Seamless Communication**
Embedding collaboration tools directly into the platform so that updates, discussions, and files live alongside the tasks and projects they relate to.
- **Scalable Security & Administration**
Providing enterprise-grade access control, compliance features, and integration management from the very start — not as an afterthought.

By filling these market gaps, Projectra aims to become the **go-to solution** for organizations seeking an adaptable, intuitive, and fully integrated project management platform.

Aims and Objectives

Aims

The primary aim of Projectra is to develop a **scalable, intuitive, and customizable project management platform** that empowers organizations to plan, execute, and monitor projects more effectively. Unlike traditional tools that either overwhelm or underserve, Projectra will deliver **balanced functionality** that adapts to the needs of diverse industries — from startups and SMEs to large enterprises.

By centralizing **Member Management, Project Management, User Management, Task Management, and App Management** into a single cohesive system, Projectra seeks to:

- Enhance **collaboration** and **transparency** across teams.
- Reduce operational inefficiencies caused by fragmented tools.
- Provide **data-driven insights** for better decision-making.
- Enable organizations to **scale their operations** without compromising workflow quality.

Ultimately, Projectra aims to **humanize project management** by making it accessible, engaging, and adaptable — ensuring that teams spend less time managing tools and more time delivering results.

Objectives

To achieve the above aim, Projectra will pursue the following **specific objectives**:

1. Member Management

- Build a robust module to **add, update, and remove** members.
- Implement **role and permission assignments** to control access to different functionalities.
- Enable grouping of members into **teams or departments** for easier management.

2. Project Management

- Allow creation of **multiple concurrent projects** with defined goals, milestones, and deadlines.
- Provide **visual tracking tools** such as Gantt charts, Kanban boards, and milestone timelines.
- Integrate **project health indicators** (e.g., progress percentage, task completion rate).

3. User Management

- Offer **role-based access control** to maintain system security.
- Implement **multi-factor authentication** for secure logins.
- Provide a **user profile system** with activity logs for accountability.

4. Task Management

- Enable creation, assignment, and prioritization of tasks with **deadlines and dependencies**.
- Allow **task-specific discussions** with attachments and comments.
- Provide **real-time notifications** for status changes and updates.

5. App Management

- Centralize **organization-wide settings**, including branding, subscription management, and integration controls.
- Provide **analytics dashboards** with usage statistics, adoption rates, and productivity insights.
- Ensure **compliance and data security controls** to meet industry regulations.

System Overview

System Introduction

Projectra is a **cloud-based, multi-platform project management application** designed to unify the essential components of team collaboration into a single ecosystem. Built using the **MERN stack** (MongoDB, Express.js, React.js, Node.js), the platform offers a **responsive, scalable, and user-friendly interface** that can be accessed from web browsers and mobile devices.

The system is divided into **five core functional modules**:

1. **Member Management** – Enables the creation, grouping, and role assignment of team members.
2. **Project Management** – Provides tools for planning, tracking, and analyzing projects.
3. **User Management** – Controls access rights, authentication, and account settings.
4. **Task Management** – Facilitates the assignment, tracking, and prioritization of individual tasks.
5. **App Management** – Allows administrators to control platform settings, integrations, analytics, and compliance.

These modules are **fully integrated**, ensuring that actions in one area (e.g., task completion) are reflected across the platform (e.g., project progress, member workload).

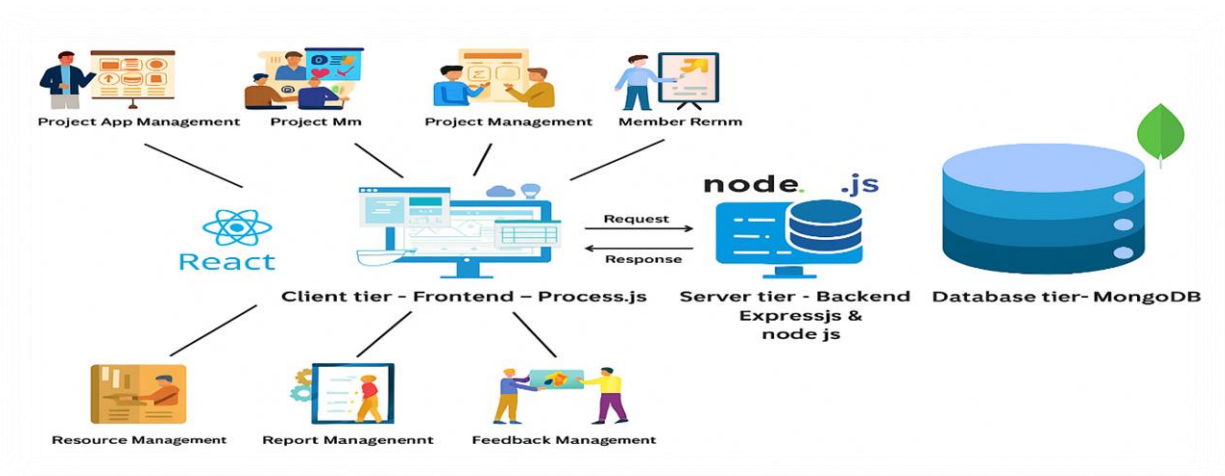


Figure 1 System Diagram

Key Features

- **Role-Based Dashboards** – Interfaces adapt to the user’s role (e.g., Admins see organization-wide analytics, Project Managers see project overviews, Team Members see their assigned tasks).
- **Real-Time Collaboration** – Task comments, file sharing, and live updates ensure everyone stays informed.
- **Customizable Workflows** – Teams can configure statuses, approval processes, and project templates to match their unique needs.
- **Cross-Platform Access** – The platform is optimized for desktop, tablet, and mobile use.
- **Integrations** – API-based integrations with tools like Slack, Google Workspace, and Microsoft Teams for extended functionality.
- **Analytics & Reporting** – In-depth dashboards for tracking productivity, resource allocation, and deadline adherence.

The architecture follows a **modular, service-oriented design** to support scalability and flexibility:

- **Frontend** – Built with React.js and Tailwind CSS for a dynamic, responsive interface.
- **Backend** – Powered by Node.js with Express.js for handling requests, authentication, and business logic.
- **Database** – MongoDB stores structured and semi-structured data such as users, projects, tasks, and logs.
- **APIs** – RESTful APIs handle communication between the frontend and backend, and between Projectra and third-party tools.
- **Security Layer** – Implements HTTPS, JWT-based authentication, and role-based access control.
- **Cloud Hosting** – Deployed on AWS for scalability, with automatic backups and load balancing.

Functional Requirements

The following functional requirements describe the essential capabilities of Projectra, categorized into the five core modules. Each module is interconnected, ensuring seamless workflow across the platform.

1 Member Management

The Member Management module enables administrators and project managers to **add, manage, and organize members** within the platform.

Key Features:

- **Member Onboarding:**
 - Add members individually or through bulk import (CSV/Excel).
 - Assign default roles (e.g., Admin, Project Manager, Contributor) during onboarding.
- **Team Grouping:**
 - Create groups for specific departments, projects, or skill sets.
 - Assign multiple members to one or more groups.
- **Role & Permission Management:**
 - Define access levels to ensure data security.
 - Modify roles dynamically as team responsibilities change.
- **Member Status Tracking:**
 - View active, inactive, and suspended members.
 - Send reminders to inactive users for re-engagement.
- **Search & Filter:**
 - Locate members by role, department, or project involvement.

2 Project Management

The Project Management module provides a **centralized workspace** for planning, executing, and tracking projects.

Key Features:

- **Project Creation:**
 - Define project name, description, start/end dates, milestones, and goals.
 - Choose from pre-defined templates for common workflows.
- **Milestone & Timeline Tracking:**
 - Set key deliverables with deadlines.
 - Visualize timelines with **Gantt charts** and milestone markers.
- **Project Views:**
 - **Kanban Board:** Organize tasks into stages (To Do, In Progress, Completed).
 - **List View:** Compact display of all tasks and deadlines.
 - **Calendar View:** Monthly/weekly scheduling for deliverables.
- **Progress Indicators:**
 - Automatic calculation of completion percentage.
 - Highlight overdue milestones for immediate attention.
- **Project Roles:**
 - Assign project-specific roles separate from global user roles.
- **Reporting:**
 - Generate detailed reports on budget utilization, resource allocation, and task completion rates.

3 User Management

The User Management module controls **authentication, access rights, and account settings**.

Key Features:

- **Authentication & Security:**
 - Email/password login, Single Sign-On (SSO), and Multi-Factor Authentication (MFA).
 - Password recovery and reset options.
- **Role-Based Access Control (RBAC):**
 - Define and manage system-wide user roles with corresponding permissions.
- **Profile Management:**
 - Allow users to update their name, contact details, and profile picture.
 - Display activity history for transparency.
- **Login History:**
 - Track login attempts and session history for auditing purposes.
- **User Status:**
 - Suspend or reactivate user accounts as needed.

4 Task Management

The Task Management module allows users to **create, assign, and monitor tasks** with full transparency.

Key Features:

- **Task Creation & Assignment:**
 - Add task details including title, description, attachments, and due dates.
 - Assign tasks to one or multiple members.
- **Task Dependencies:**
 - Link tasks that depend on others for completion.
 - Automatically update dependent tasks when preceding tasks are completed.
- **Priority Levels:**
 - Assign priority (Low, Medium, High, Critical) to guide resource allocation.
- **Collaboration Tools:**
 - Comment threads within each task.
 - File uploads and document linking.
- **Progress Tracking:**
 - Visual progress bar per task.
 - Task aging alerts for overdue assignments.
- **Task Views:**
 - List, Kanban, and Calendar views for flexibility.

5 App Management

The App Management module offers **administrative control** over the platform's settings, integrations, and analytics.

Key Features:

- **Organization Settings:**
 - Configure branding (logo, color scheme) for a personalized interface.
 - Manage subscription plans and billing details.
- **Integration Management:**
 - Connect with third-party tools (Slack, Google Workspace, Microsoft Teams).
 - Enable/disable integrations based on company policy.
- **Platform Analytics:**
 - Monitor system usage, feature adoption, and team activity trends.
- **Compliance & Security:**
 - Access audit logs for all system changes.
 - Ensure GDPR and data privacy compliance.
- **Backup & Recovery:**
 - Automated daily backups with one-click restore functionality.

1. Feedback Management

The feedback management system enables team members and stakeholders to submit reviews and share their experiences regarding tasks and project progress. Users can also view feedback from others to gain insights and improve collaboration. Project managers have access to all submitted feedback and can respond to specific comments, ensuring timely resolution of concerns. By analyzing feedback trends, project teams can enhance workflow efficiency, improve communication, and address recurring issues effectively.

2. Report Management

The reporting system provides project managers with insights into project milestones, task completion rates, and resource utilization over daily, monthly, and yearly periods. They can generate detailed reports for specified date ranges to support project analysis and planning. The system also offers print-friendly formats, facilitating easy documentation and record-keeping for project reviews and performance evaluations.

Nonfunctional Requirements

Nonfunctional requirements define the **quality attributes** and constraints of the Projectra platform. These ensure that the system not only works as intended but also meets user expectations in terms of **performance, security, scalability, usability, and maintainability**.

1 Performance

- The system must support **at least 500 concurrent active users** during initial deployment, with the ability to scale to **5,000+ concurrent users** without significant degradation in response time.
- Average **page load time** should not exceed **3 seconds** under normal operating conditions.
- **Real-time updates** (e.g., task status changes, chat messages) should be reflected across all connected clients within **2 seconds**.
- The backend must be able to handle **a minimum of 100 API requests per second** during peak hours without failure.

2 Scalability

- The architecture should be **modular and service-oriented**, allowing individual components (e.g., Task Management, Reporting) to scale independently.
- The database should be designed for **horizontal scaling**, supporting sharding and replication for large datasets.
- Cloud infrastructure must allow **auto-scaling** to handle unexpected spikes in user activity.
- The system should be capable of integrating **additional modules and features** without requiring major refactoring.

3 Availability

- Projectra must maintain a **99.5% uptime** target during its first year of operation, with a long-term goal of **99.9% uptime**.
- **Scheduled maintenance windows** must be communicated to users at least **48 hours in advance**.
- **Failover mechanisms** must be in place so that critical functions (e.g., task updates, communication tools) remain operational during partial outages.
- **Automated daily backups** must be performed to ensure recovery in the event of system failure.

4 Usability

- The user interface must follow **responsive design principles** to ensure seamless use across desktops, tablets, and mobile devices.
- The system should be **intuitive enough** for new users to begin managing projects with minimal onboarding (no more than **30 minutes** of training).
- **Accessibility compliance** (WCAG 2.1 AA) should be met to ensure usability for people with disabilities.
- All primary actions (e.g., creating a task, assigning a member) should require **no more than three clicks** from the relevant dashboard.

5 Security

- All sensitive data (e.g., passwords, tokens) must be **encrypted** at rest using AES-256 and in transit using TLS 1.3.
- **Role-Based Access Control (RBAC)** must be enforced at both frontend and backend levels.
- **Multi-Factor Authentication (MFA)** should be available for all accounts.
- The platform must comply with **data protection regulations** (e.g., GDPR, CCPA) relevant to the user's region.
- Regular **penetration testing** must be conducted to identify and patch vulnerabilities.

6 Maintainability

- The codebase should be **fully documented** to facilitate onboarding of new developers.
- The system should support **continuous integration and continuous deployment (CI/CD)** for rapid updates.
- Automated test suites should ensure **at least 80% code coverage** for backend and frontend components.
- All dependencies should be **regularly updated** to minimize security risks.

Technical Requirements

Frontend

- **Framework:** React.js for building a dynamic, responsive, and interactive user interface for project, task, and team management.
- **Styling:** Tailwind CSS for creating a modern, mobile-friendly UI that adapts to various screen sizes.

- **Compatibility:** Cross-browser support (Chrome, Firefox, Safari, Edge) with a responsive layout for desktop, tablet, and mobile devices.

Backend

- **Framework:** Node.js with Express.js for handling server-side logic, API endpoints, authentication, and data processing.

Database

- **Type:** MongoDB (NoSQL) for flexible and scalable data storage.
- **Collections:**
 - **Users:** Stores user profiles, authentication data, roles (e.g., admin, manager, member).
 - **Projects:** Stores project details, timelines, and assigned members.
 - **Tasks:** Stores task descriptions, priorities, deadlines, and status updates.
 - **Teams:** Stores member assignments, roles, and permissions.
 - **Feedback:** Stores user and client feedback for system improvements.
 - **Reports:** Stores analytics data for productivity, task completion rates, and project health.

APIs

- **RESTful APIs** for communication between frontend and backend.
- **Third-party APIs** for:
 - **Authentication:** Secure user sign-in (e.g., OAuth, JWT).
 - **Notifications:** Real-time task and project status updates.
 - **File Storage:** Integration with services like Cloudinary or AWS S3 to store project files and documents.
 - **Chat & Collaboration:** Integration with messaging APIs for real-time member communication.

Hosting & Deployment

- **Hosting:** Cloud-based deployment (e.g., Vercel, AWS, or Render) for scalability and high availability.

Version Control

- **Git & GitHub** for tracking changes, managing branches, and enabling collaborative development.

Other Features & Tools

- **Search and Filter:** Fast search for tasks, projects, and members with multiple filtering options.
- **Role-Based Access Control:** Restricts access to sensitive project data based on user roles.
- **Real-Time Dashboard:** Live updates for task progress, deadlines, and resource allocation.
- **Reporting & Analytics:** Automatic report generation for performance, deadlines, and workload distribution.

Literature Review

To design and develop the proposed Project Management System, we reviewed several widely used project and task management platforms, including **Trello**, **Asana**, **ClickUp**, and **Jira**. The objective was to compare their key features, identify strengths and weaknesses, and determine how our system could provide similar or improved functionalities for efficient project execution.

User Account & Role Management

All reviewed platforms support account creation and allow administrators to manage roles and permissions. Users can be assigned as admins, project managers, or members with varying access levels. Our system adopts a similar structure, using **role-based access control** to secure sensitive project data while enabling collaboration between members.

Project & Task Management

Tools like Trello and Asana use boards, lists, and cards for visual project management, while Jira focuses on Agile workflows such as Scrum and Kanban. We aim to combine these approaches by

offering **task boards, Gantt charts, and list views**, giving users flexibility in how they view and manage projects.

Collaboration & Communication

ClickUp and Jira integrate chat and comment threads directly into tasks, enabling better communication. Our system will offer **built-in chat and file-sharing features**, ensuring that members can collaborate without needing external communication tools.

Reporting & Analytics

Jira and Asana provide powerful analytics, tracking task completion rates, workload distribution, and progress over time. We will incorporate **real-time analytics dashboards and automated report generation**, helping managers make data-driven decisions.

Search, Filter, and Notifications

All competitor systems include advanced search, filtering, and notification capabilities. Our platform will enhance this with **real-time notifications** for task updates, deadlines, and new assignments, plus **multi-criteria filtering** for tasks and projects.

Integration & Extensibility

Most reviewed platforms integrate with third-party tools like Google Drive, Slack, and GitHub. Our system will provide **API integrations for file storage, authentication, and productivity tools** to ensure adaptability to different organizational workflows.

Security & Data Privacy

Competitors implement secure authentication and encryption for user data. Our system will follow industry standards with **HTTPS encryption, hashed passwords, and secure API endpoints**, ensuring data privacy and compliance.

By evaluating these existing tools, we identified opportunities to merge **ease of use** with **powerful analytics, real-time collaboration, and customizable workflows** — creating a system that not only competes with but also improves upon existing project management solutions.


Feature	Trello	Asana	Jira	ClickUp	 Our System
User Account Management	✓	✓	✓	✓	✓ (Role-based access control)
Project Management	✓	✓	✓	✓	✓ (Multiple secure options options)
Task Management	✓	✓	✓	✓	✓ (Priority-less cause deadlines, dependencies)
Activity History	✓	✓	✓	✓	✓ (Quick search & filter for past activities)
Task Creation & Editing	✓	✓	✓	✓	✓ (Bulk creation & quick edit options)
Real-Time Collaboration	✓	✓	✓	✓	✓ (Live updates & chat integration)
Notifications & Reminders	✓	✓	✓	✓	✓ (Customizable alerts & deadlines)
Reporting & Analytics	Basic	(Basic)	(Advanced)	(Advanced)	✓ (Custom report builder & export)
Search & Filter	✓	Check	✓	✓	✓ (Customizable alerts & expertise)
Integrations	Limited	Popular tools	Dev tools	Variety	✓ (Custom API & external integrations)
File Sharing	✓	✓	Advanced	✓	✓ (Unlimited file uploads with tagging)
Mobile App Support	Basic	Advanced	Advanced	Advanced	✓ (Read & edit mode offline)
Custom Workflows	Basic	Advanced	Advanced	Advanced	✓ (Optimized for all devices)

Figure 2 Comparison with similar products

Methodology

Requirements Engineering Methods

To ensure a complete understanding of both organizational needs and end-user expectations, a range of requirements engineering techniques were used. These techniques supported the development of a functional and scalable Project Management System and provided valuable insights into current operational challenges.

- **Stakeholder Interviews:** Structured interviews were conducted with project managers, team leaders, and potential system users to gather detailed information about project tracking needs, collaboration challenges, and required functionalities. These discussions helped prioritize core features and identify weaknesses in existing manual or semi-digital processes.
- **Observation of Current Workflow:** Direct observation sessions were carried out within the current project management environment to study task assignment, communication methods, and reporting workflows. This approach provided real-world insights into inefficiencies, delays, and recurring issues that interviews alone might not fully capture.

Alternative Considered: A survey was considered to collect information from a larger group of potential users. However, to gain deeper and more targeted insights into workflow requirements, interviews and direct observation were preferred.

Design Methods

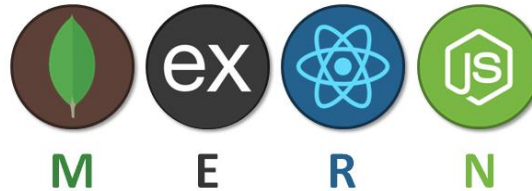
- **Wireframing with Visily:** Used to design the user interface for dashboards, project boards, and task views. This ensured a clean, intuitive, and user-friendly layout before development began.
- **ERDs with Draw.io:** Entity-Relationship Diagrams were created to define the database schema, ensuring clear relationships between entities such as Users, Projects, Tasks, Teams, and Reports.
- **System Diagram with Canva:** A high-level system diagram was developed to visualize the architecture of the Project Management System, showing interactions between the frontend, backend, and database.
- **DFD with Visual Paradigm:** Data Flow Diagrams were used to map how data moves within the system—from project creation to task updates, reporting, and analytics—ensuring efficient data handling.
- **Flowchart with Visily:** Created to illustrate user interactions and process flows, including task creation, assignment, and status updates.
- **Onion Diagram with Draw.io:** Used to define the layers of the system's architecture, ensuring a clear separation of concerns between UI, business logic, and data storage.

Alternative Considered: A fully interactive prototype was considered to provide high-fidelity user testing before development. However, due to time constraints in the startup phase, wireframes and diagrams were prioritized for faster iteration and implementation.

Tools and Technologies

We use MERN stack to develop our web application.

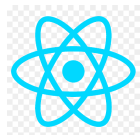
: MongoDB, Express JS, React JS, and Node JS.



A NoSQL database that stores data in JSON-like format, making it flexible and scalable. It's ideal for handling large volumes of unstructured data (MongoDB, Inc., 2025). <https://www.mongodb.com/>



A lightweight web application framework for Node.js, used for building robust and scalable backend applications. It simplifies routing and middleware handling (Express.js, 2025). <https://expressjs.com/>



A JavaScript library for building user interfaces, especially single-page applications. React enables the creation of dynamic, responsive UIs with reusable components (React, 2025). <https://reactjs.org/>



A JavaScript runtime built on Chrome's V8 engine, allowing the development of server-side applications in JavaScript. It's asynchronous and event-driven, making it efficient for handling I/O-heavy tasks (Node.js Foundation, 2025). <https://nodejs.org/>

Apart from MERN Stack we use following tools and technologies in our project.



VS Code is a free, open-source code editor developed by Microsoft. It provides support for multiple programming languages, a powerful debugging environment, and a wide range of extensions for productivity. VS Code is favored for web development due to its versatility, lightweight design, and robust feature set (Microsoft, 2025). <https://code.visualstudio.com/>



GitHub is a platform for version control and collaborative software development. It uses Git to track changes in code and provides tools for managing repositories, branching, pull requests, and collaboration among developers. GitHub is widely used for open-source projects and private development (GitHub, Inc., 2025). <https://github.com/>



Tailwind CSS is a utility-first CSS framework that allows developers to build custom designs quickly. Instead of writing custom styles, developers apply pre-defined utility classes directly to HTML elements. This results in highly customizable, responsive designs with minimal CSS code (Tailwind Labs, 2025). <https://tailwindcss.com/>



ApexCharts is a modern charting library that enables developers to create interactive and responsive charts for web applications. It supports a wide range of chart types, including line, bar, pie, and area charts, allowing for detailed data visualization. ApexCharts offers easy integration with popular frameworks and libraries like React, Angular, and Vue, ensuring seamless embedding of charts to enhance user experience (ApexCharts, 2025). <https://apexcharts.com/>

Integration Methods

- **RESTful APIs:** Use for communication between the frontend and backend. RESTful APIs allowed for efficient and scalable data exchange, where each request is processed

based on standard HTTP methods (GET, POST, PUT, DELETE) to handle different resources.

- **ApexCharts:** used to create interactive and visually appealing data visualizations. It provides a reliable and easy-to-integrate solution for embedding charts within the app, ensuring users can easily understand complex data through clear and responsive graphical representations.

Agile Software Engineering Methodology



Figure 3 Agile Methodology

Agile is a kind of software development process that applies a certain amount of pragmatism to the final product delivery and foresees the requirement for flexibility. Because agile software development emphasizes the clean delivery of discrete components rather than the full product, it necessitates a cultural shift in many organizations.

- **People and interactions over procedures and tools:** This principle stresses the value of team members communicating and working together rather than depending only on procedures and tools.

- Working software above thorough documentation: Agile teams place a higher priority on working software than thorough documentation because they consider software to be the most crucial product.
- Agile teams place a higher priority on client collaboration than contract negotiation in order to make sure that the product they are creating satisfies their demands and specifications.
- Adapting to change rather than sticking to a plan: Agile teams understand that requirements can change over time and that software development is an iterative process. When reacting to these developments, they place a high value on adaptation and flexibility.

Software development teams frequently employ the agile technique because it enables them to respond more quickly to shifting priorities and requirements. Agile teams prioritize and finish a set of activities or user stories in brief iterations, or sprints, that usually run one to four weeks, as opposed to adhering to a strict schedule. The goal of each sprint is to produce a functional software increment that clients can test and certify by organizing, carrying out, and evaluating a series of development tasks.

In order to make sure that everyone is working towards the same objectives and that the final product satisfies the demands of the customer, agile teams also place a strong emphasis on regular communication and collaboration between team members, including the customer or end user.

Motives behind the adoption of Agile Methodology

Client satisfaction: The goal of agile methodology is to continuously and swiftly provide value to the client. This makes it more likely that the product will satisfy the needs of the client and arrive on schedule.

Flexibility: Agile methodology can be easily adjusted to meet shifting priorities and requirements. This enables teams to react swiftly to modifications in the project's scope or the state of the market. **Cooperation:** The agile technique places a strong emphasis on teamwork,

communication, and cooperation. This can result in better solutions and foster a sense of shared accountability and ownership.

Transparency: Throughout the development process, open communication and transparency are promoted by agile methods. This guarantees that everyone is working towards the same objectives and fosters trust between team members and stakeholders.

Continuous improvement: Agile methodology places a strong emphasis on ongoing education and development. This makes it possible to guarantee that the group is continuously improving its procedures and methods to produce better outcomes.

Criteria	Agile	Waterfall	Spiral Model
Flexibility & Adaptability	✓ Agile is highly adaptable, allowing easy adjustments based on evolving client needs or market conditions.	✗ Waterfall follows a rigid structure, making it difficult to accommodate changes once the project starts.	✓ Spiral allows for flexibility, enabling iterations and refinements based on regular risk assessments and client feedback.
Client Involvement	✓ Continuous client feedback ensures that the product aligns with expectations, reducing misalignment.	✗ Limited client involvement until later phases, making it harder to meet changing requirements.	✓ Regular client involvement is part of the Spiral process, especially during each iteration to ensure the product aligns with their needs.
Iterative Development	✓ Develops the product in small, manageable iterations, allowing for continuous testing and refinement.	✗ Development happens in a single, long phase, which delays feedback and adaptation.	✓ Spiral uses iterations (phases), allowing for constant refinement, validation, and adjustments throughout the project.
Time to Market	✓ Agile enables quicker releases and faster feedback, helping deliver a functional product sooner.	✗ The linear, sequential process delays time-to-market and may result in late-stage issues.	✓ Spiral allows for faster delivery of incremental product features through multiple iterations.
Risk Management	✓ Regular testing and feedback reduce risks early by identifying issues early in development.	✗ Risk management is delayed until later stages, where it may be harder to make corrections.	✓ Spiral includes a dedicated focus on risk management in each iteration, ensuring early detection and mitigation of risks.
Continuous Improvement	✓ Agile fosters a culture of continuous learning and	✗ Waterfall lacks opportunities for ongoing improvement,	✓ Spiral allows for continuous improvement in each iteration, with opportunities to reassess

	improvement throughout the project.	as it follows a strict plan from start to finish.	and improve the product after each cycle.
Clear Requirements at the Start	✗ Agile embraces change, so requirements are refined over time rather than being fixed upfront.	✓ Waterfall requires clear, well-defined requirements at the beginning, which may be unrealistic for dynamic projects.	✓ Spiral allows for evolving requirements and refines them during each iteration, making it adaptable to changing project needs.
Suitable for Large, Complex Projects	✗ While suitable for many projects, Agile works best with smaller, more flexible projects.	✓ Waterfall is better suited for larger, complex projects with stable requirements.	✓ Spiral is ideal for large, complex projects, as it allows for ongoing risk management and multiple iterations of feedback and development.
Predictable Timeline	✗ Agile may involve changes to timelines due to its adaptive nature.	✓ Waterfall provides a clear, predictable timeline, ideal for projects with strict deadlines.	✓ Spiral has a predictable iterative cycle but can extend based on risk evaluations and project complexity.
Structured Approach	✗ Agile lacks a strict, rigid structure but is flexible in accommodating changes.	✓ Waterfall follows a detailed, structured approach that is easy to follow.	✓ Spiral has a well-defined structure for managing risks and iterations, making it systematic, but still flexible.

Conclusion:

Agile remains the most suitable methodology for the **Innovation-Based Project Management App**. It provides the flexibility, rapid iteration, and continuous client and stakeholder feedback essential for developing a dynamic and user-focused application. Given the likelihood of evolving requirements and frequent updates, Agile's iterative approach supports quick adaptation and ongoing improvement.

The Spiral Model is also a viable alternative, especially for larger or more complex project phases that require thorough risk management and iterative refinement. However, while Spiral offers flexibility, its complexity can make it harder to manage compared to Agile and may not facilitate as frequent client interaction and fast delivery cycles.

Waterfall is the least appropriate for this project due to its rigid structure and lack of ongoing collaboration, which limits responsiveness to change and reduces opportunities for continuous stakeholder engagement.

Therefore, Agile stands out as the most appropriate methodology for the development of this project management app, given its client-centric approach, iterative progress, and emphasis on fast, adaptive delivery.

Project plan (Gantt chart)

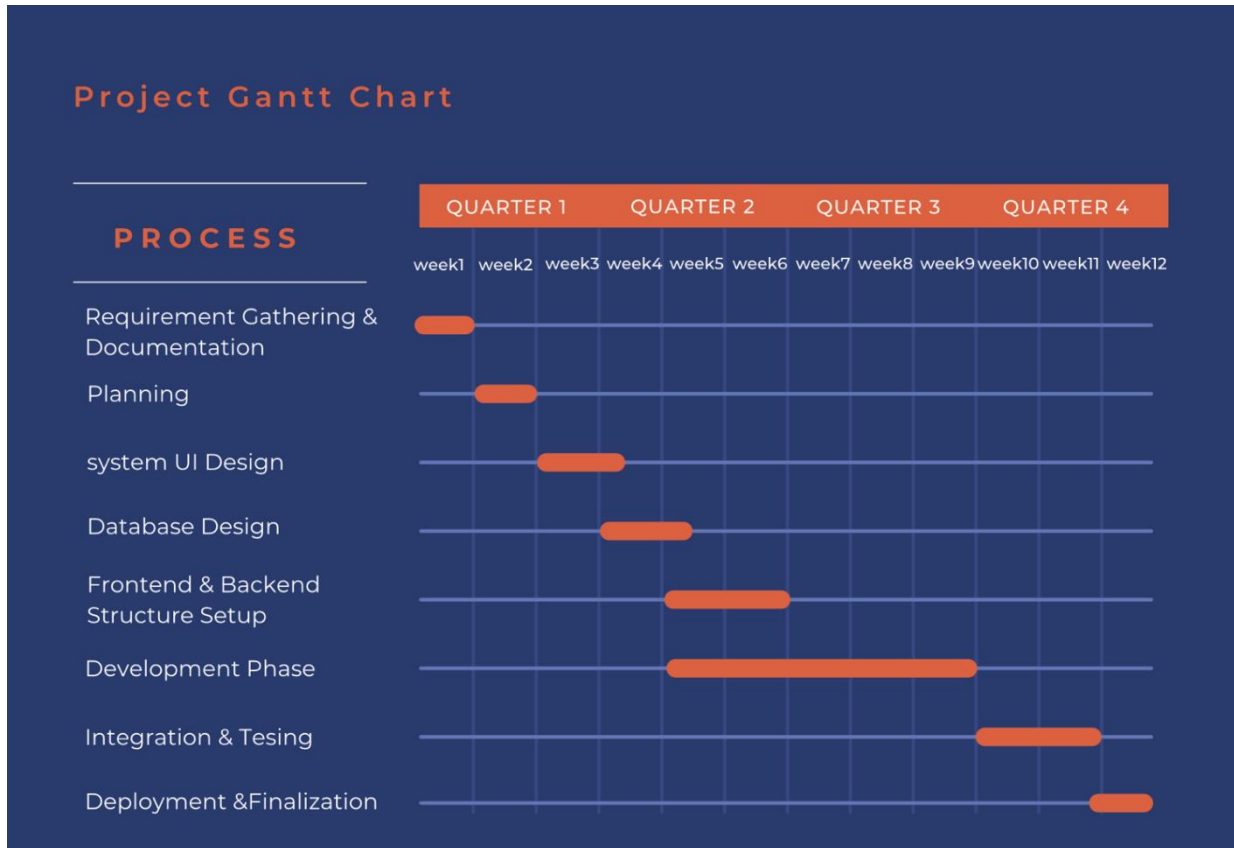


Figure 4 Gantt chart

We started the requirement gathering & documentation in the first week of semester. We could gather and document all necessary information from the client through interviewing in the first week, based on the requirements we planned the system to meet the client's needs. In the third week we have decided to design the user interfaces of the system (wireframes).

We decided to begin the database design (Data Modeling, Collections and Documents, ER Diagram, Normalization vs. Denormalization, Data Validation, Data Relationships, Backup and Recovery Plan, Security and Access Control) in the fourth week and expect to finish it mid of the fifth week.

Parallely, frontend and backend structure setup and the development will be started in the fifth week, we expect the development process will be finished before the tenth week.

After completed the testing will be conducted, if anything wants to change it will happen from week 10 to end of week 11. (unit testing, integration testing, user acceptance testing). The test manually will be conducted by us and the client.

After finalizations, we planned the deployment on the 12th week.

Work breakdown structure (Work distribution)

Name with Initials and student ID	Task Description	Additional Tasks Completed
K.Seyon IT2375272	<p>Establishment of Project App Management: Develop a system to manage the complete lifecycle of projects, from creation to final delivery. This includes tracking progress, setting deadlines, assigning deliverables, and ensuring timely completion. The system will provide real-time updates for both clients and team members, improving workflow efficiency and transparency.</p> <p>Establishment of Manual Task Entry for Non-Technical Users: Create an interface where team members can manually enter tasks into the system, which are then reviewed and approved by the project manager for accuracy before scheduling.</p> <p>Feedback System: Implement a feedback system that allows team members and project stakeholders to provide ratings and reviews on tasks, milestones, and overall project progress. This will help project managers assess team performance, identify issues early, and enhance collaboration and productivity.</p> <p>Reporting System: Develop a reporting system that generates detailed reports on project status, resource allocation, task completion rates, and team feedback. This will enable project managers to analyze trends, optimize workflows, and make informed, data-driven decisions to improve project outcomes.</p>	Analyze Company Background, Analyze the project Aim and Objective

<p>K.Braveenraj IT23204416</p>	<p>Development of Centralized Project & Task Management Dashboard: Build an intuitive interface for adding, editing, categorizing, and tracking projects and tasks. The dashboard should allow filtering by department, priority, or deadline, improving project visibility. Managers will have backend access to ensure all data remains accurate.</p> <p>Integration of AI ChatBot for Project Assistance: Add a ChatBot that assists users in finding project-related information, answering FAQs, suggesting next steps, and providing instant status updates.</p>	<p>Analyze the functional, NFRs. And technical requirements. Development of system diagram.</p>
<p>N.Vinushan IT22545862</p>	<p>User Management :Develop a comprehensive User Management system focused on providing seamless and secure access for all app users. The system will allow users to create and manage their accounts easily, enabling personalized experiences such as customized dashboards, role-based access, and exclusive project-related notifications. Users can update their profiles, manage preferences, and receive real-time updates on task assignments, project progress, and team communications. The User Management function aims to enhance user convenience and improve overall project collaboration by ensuring smooth, efficient, and user-friendly account handling within the app.</p>	<p>Analyze Company Background. Analyze the project problems and Motivations.</p>
<p>P.Sinthujan IT2385620</p>	<p>User Account and Task Tracking System:</p> <p>Create a seamless and convenient user management system that allows team members and project stakeholders to create accounts by providing essential personal details (e.g., name, email address, contact number, and password). After logging in, users can access project tasks, assign responsibilities, and select task priorities.</p> <p>The system will validate task assignments to ensure workload balance within the team. If a user is overloaded or unavailable, task assignment will be restricted to maintain productivity. Multiple notification options (e.g., email alerts, in-app notifications) will be available.</p> <p>A short cancellation or reassignment window will be provided, allowing users to update or cancel task assignments within a limited timeframe without penalty. Once tasks are confirmed, the</p>	<p>Analyze and compare the existing system. Analyze the suitable for the project methodology.</p>

	<p>system will provide real-time tracking and status updates such as "Task Assigned," "In Progress," and "Completed."</p> <p>Additionally, users can track project milestones and progress through live updates. Account management features will allow users to view their task history, update personal information, and manage account settings to ensure smooth project collaboration and management.</p>	
S.Rishikeshan IT23778344	<p>Member Management System: Develop a system to efficiently manage project team members by tracking their roles, availability, and task assignments. The system will automatically update member status based on task progress and notify project managers when members are overloaded or underutilized. It will also allow easy addition, removal, or reassignment of team members to ensure optimal resource allocation and smooth project execution.</p>	Tool and technology analysis

Evaluation Method

Ensuring the effectiveness, reliability, and overall quality of the Online Food Ordering System is critical to its successful deployment and long-term usability. This section details the comprehensive evaluation strategy designed to assess all aspects of the system's functionality, performance, security, and user experience.

Evaluation Criteria

The evaluation will be guided by a set of clearly defined criteria to ensure comprehensive coverage of all quality aspects:

- **Functionality:** Verifies that all features operate according to the requirements specification, ensuring full feature completeness and correctness.
- **Usability:** Measures the user-friendliness and intuitiveness of the system interface through structured user surveys and observation studies.

- **Performance:** Evaluates system responsiveness and stability under varying workloads to ensure optimal performance during peak usage.
- **Security:** Assesses the system's capability to protect sensitive information against unauthorized access and cyber threats.
- **Reliability:** Ensures the system consistently performs as expected without unexpected failures or downtime.
- **Maintainability:** Reviews the system's codebase and architecture to determine ease of future modifications, debugging, and enhancements.

Testing Methods

Testing is an integral part of ensuring the quality, performance, and security of the application. A multi-layered approach was employed to validate different aspects of the system. The following testing methods were utilized.

- **Unit Testing (Jest):** Used for validating individual components in isolation to ensure each function or class works as expected. Jest was chosen for testing React components in the frontend and backend logic in Node.js, allowing for early detection of errors.
- **Integration Testing (Postman):** Focused on testing the interaction between system components. Postman was employed to validate the backend API by sending requests to the server and ensuring correct data handling and response.

Alternative Considered: While manual testing was initially planned, it was ultimately deemed insufficient for thorough validation due to the complexity of testing various system components and interactions. Automated testing tools were chosen for their efficiency and coverage.

Reference

- [1] **Trello**, [Online]. Available: <https://trello.com/>.

- [2] **Asana**, [Online]. Available: <https://asana.com/>.

- [3] **Jira**, [Online]. Available: <https://www.atlassian.com/software/jira>.

- [4] **Monday.com**, [Online]. Available: <https://monday.com/>.

Appendix

Figure 1 System Diagram 09

A visual representation of a system, depicting its components, their interactions, and the flow of information between them, essentially providing a graphical overview of how a system functions and is structured, often used to understand, design, analyze, and document complex processes or systems.

Figure 2 Comparison with similar products..... 19

A summary of existing products on project management app, where all the features are compared and discussed briefly, highlights the key functionalities, strengths, and limitations of various platforms.

Figure 3 Agile Methodology 23

Agile methodology is a project management approach that emphasizes flexibility, collaboration, and continuous improvement by breaking projects into small, manageable cycles.

Figure 4 Gantt chart 27

A visual representation, used to plan and track the progress of a project by displaying a timeline of tasks, their start and end weeks, allowing members to easily see the overall schedule.