

TaskFlow

PROJECT PLAN DOCUMENTATION

**Pericles Lelos, Krish Mohan
Sam Fickle, Charles Timmons**

Table of Contents

1. Introduction.....	
1.1 Overview.....	
2. Project Organization.....	
2.1 Team Structure.....	
2.2 Team Responsibilities.....	
3. Assumptions and Constraints.....	
4. Risk Management.....	
5. Technical Process.....	
5.1 Tools and Methods.....	
5.2 Development Environment.....	
6. Work Breakdown Structure	

7. Project Timeline

1) Introduction

1.1 Overview

Our team aims to create a web-based application with the capability of tracking tasks for the users. The intention of this application is to create an easy way for people to stay on top of tasks or chores and ensure deadlines are met using a priority system implemented in the software. This application can be applicable to anyone, but would be especially beneficial to students to ensure assignments and projects are turned in on time.

Our team has defined this project's goals to be:

1. Provides easy and reliable authorization and authentication for users
2. Develops an interface that allows users to create new task cards and assign
3. Automatically assigns priority to each task based on factors like due date
4. Develops database to store user login info and user created tasks

A successful development is achieved by meeting the project goals as specified. Specific use cases and features are listed out in the Requirements Specification Document.

2) Project Organization

2.1 Team Structure

Our team is structured democratically with each member communicating with the others to meet requirements and stay on track. Each group member has taken a flexible role to fulfill requirements depending on what needs the most attention.

2.2 Team Responsibilities

Project Manager - Ensures project progress is steady and facilitates discussion on deadlines and remaining work while helping developers when necessary

Writer/Tester - Writes documentation for each step of the project and creates/executes test cases to ensure project quality while helping developers when necessary

Frontend Developer - Focus on creating the UI for the web-based application, working in React and communicating with others to collaborate on different components

Backend Developer - Focus on creating the database for user data storage and communicating with the frontend to create routes for CRUD operations

3) Assumptions and Constraints

- Our team assumes users are familiar with web site interfaces and have access to a steady internet connection.
- Our team assumes tools and process used will remain compatible throughout the development process
- This project is designed for web browsers and does not include native mobile or desktop application development.

4) Risk Management

1. **Risk:** technical inexperience in software such as React or Docker
- **Response:** Assign roles based on confidence and cater to each member's strengths
2. **Risk:** Underestimate amount of work due
- **Response:** Assign Project Manager to facilitate work schedule and ensure project is delivered on time
3. **Risk:** Connecting the front end web application to the back end database
- **Response:** Ensure communication between front end and back end developers to prioritize a connection early on

5) Technical Process

5.1 Tools and Methods

This application will use React to build the front end user interface, and draw.io for design models and system analysis. The team will use Github as a version control tool to maintain consistency in code and documentation.

5.2 Development Environment

Operating System	Microsoft Windows 11
Software Languages	Typescript, CSS
Database	Docker

6) Work Breakdown Structure (WBS)

Project – TaskFlow Task Tracking Web Application

1. Requirements & Planning

- 1.1 Review project goals
 - 1.2 Gather and document requirements
 - 1.3 Create Requirements Specification document
 - 1.4 Define project scope, assumptions, and constraints
 - 1.5 Develop project plan (roles, risks, tools, timeline)
-

2. System Design

- 2.1 Create UI wireframes and component mockups
 - 2.2 Design system architecture (frontend–backend–database flow)
 - 2.3 Define database schema (user data, tasks, priorities)
 - 2.4 Create use cases and design diagrams (draw.io)
-

3. Frontend Development (React)

- 3.1 Set up React project structure
- 3.2 Implement authentication pages (login, signup)
- 3.3 Build dashboard/home view
- 3.4 Create task card components

- 3.5 Build create/edit task interfaces
 - 3.6 Implement task list display and priority sorting
 - 3.7 Styling with Tailwind CSS
-

4. Backend Development

- 4.1 Set up Docker environment
 - 4.2 Build database tables/collections (users, tasks)
 - 4.3 Implement CRUD API for tasks
 - 4.4 Implement authentication/authorization logic
 - 4.5 Integrate backend with frontend routes
-

5. Integration

- 5.1 Connect React frontend to backend API
 - 5.2 Ensure data can be created, edited, retrieved, deleted
 - 5.3 Test database interactions
 - 5.4 Validate priority assignment workflow
-

6. Testing

- 6.1 Develop test cases based on use cases
 - 6.2 Conduct unit testing (frontend + backend)
 - 6.3 Conduct integration testing
 - 6.4 Conduct user acceptance testing
 - 6.5 Log and resolve bugs
-

7. Deployment

- 7.1 Prepare final Docker environment
 - 7.2 Deploy application locally or to a hosting environment
 - 7.3 Verify deployment functionality
 - 7.4 Final documentation for deployment steps
-

8. Final Documentation & Presentation

- 8.1 Update Requirements Specification document
- 8.2 Update Project Plan with completed sections
- 8.3 Prepare project presentation/demo
- 8.4 Submit final project deliverables

7) Project Timeline

The development schedule for the TaskFlow application uses the unified process, where All workflows are performed over the entire life cycle. However, at most times one workflow predominates.

Week 1 – Requirements & Design

- Finalize project requirements and scope
- Complete Requirements Specification document
- Create UI wireframes and system architecture diagrams
- Define database schema (users, tasks, priority)

Week 2 – Frontend Foundations

- Set up React project environment
- Implement login and sign-up pages
- Begin creating task UI components (task card, forms)
- Set up Tailwind CSS styling framework

Week 3 – Backend Development

- Set up Docker database environment
- Build user and task database structures
- Develop backend authentication logic
- Implement CRUD routes for task operations

Week 4 – Integration & Core Features

- Connect frontend UI to backend API
- Enable creating, editing, and deleting tasks
- Retrieve and display user tasks with priority sorting
- Begin integration testing

Week 5 – Testing, Fixes, and Final Delivery

- Conduct full testing (unit, integration, and acceptance)
- Fix bugs and polish UI
- Prepare final documentation and project demo
- Complete deployment environment setup