Software Requirements Specification Ver1.0

Developer: Zhenqi Zhang

Table of Contents

[Introduction 4](#_Toc207352216)

[Purpose 4](#_Toc207352217)

[Scope 4](#_Toc207352218)

[Audience 4](#_Toc207352219)

[Functional Requirements 5](#_Toc207352220)

[Task Management 5](#_Toc207352221)

[Calendar View 5](#_Toc207352222)

[Time Tracking 5](#_Toc207352223)

[Notification 5](#_Toc207352224)

[Dashboard & Analytics 5](#_Toc207352225)

[AI-Powered Features (Optional) 6](#_Toc207352226)

[Non-Functional Requirements 7](#_Toc207352227)

[Usability 7](#_Toc207352228)

[Performance 7](#_Toc207352229)

[Reliability 7](#_Toc207352230)

[Scalability 7](#_Toc207352231)

[Security 7](#_Toc207352232)

[System Architecture 8](#_Toc207352233)

[Frontend (React) 8](#_Toc207352234)

[Backend (Express) 8](#_Toc207352235)

[Database (PostgreSQL) 8](#_Toc207352236)

[DevOps & CI/CD (Docker) 8](#_Toc207352237)

[Hosting & Development (To be conducted) 8](#_Toc207352238)

[UI/UX Design (Figma) 8](#_Toc207352239)

[Development Tools & Stack Summary 8](#_Toc207352240)

[Project Milestones 8](#_Toc207352241)

# Introduction

## Purpose

This document outlines the requirements for developing a productivity software application focused on task management, time tracking, and data analysis. The software will allow users to organize their to-do lists on daily, weekly, monthly, and yearly levels, with a calendar view, time tracking features, and intelligent suggestions based on user behaviour. The purpose of this software is to provide users with a good planning platform, allowing them to record the plans they think of at any time.

## Scope

This software aims to:

Provide a calendar-based task management system

Allowing drag-and-drop scheduling

Track time for tasks

Offer dashboard with analytics

Send helpful notification

Offer AI-based suggestions for planning.

## Audience

This document is intended for: Developers, Designers, Project Managers and Future Contributors

# Functional Requirements

## Task Management

* Create tasks with titles, descriptions, tags and priorities
* Set task as daily, weekly, monthly or yearly
* Assign due dates and time blocks
* Support recurring tasks
* Allow categorization by type (Work, Personal)

## Calendar View

* View tasks in day/week/month/year formats
* Navigate to any date via calendar
* Drag and drop tasks to schedule them
* Adjust task duration by dragging
* Color-coded task types

## Time Tracking

* Start/Stop timers for tasks
* View historical records of time spent
* Assign time logs to tasks
* Generate time reports per day/week/month

## Notification

* Remind users of upcoming tasks
* Notify users after long continuous work sessions
* Set custom reminders

## Dashboard & Analytics

* Display task completion rate
* Visualize productivity by type/category
* View time spent per task or category
* Display trends over time (Line charts, pie charts)

## AI-Powered Features (Optional)

* Daily summaries generated by AI
* Suggest daily plans based on past habits
* Smart reminders based on typical schedule

# Non-Functional Requirements

## Usability

* Clean and intuitive UI/UX
* Responsive design for desktop and tablet
* Can easily add their plans to the application

## Performance

* Fast rendering calendar and task list
* Time tracking accurate to the second
* Provide support with planning, through AI

## Reliability

* Save data locally and/or in the cloud
* Auto-save task data

## Scalability

* Able to support large number of tasks and users

## Security

* User authentication
* Data encryption for personal tasks and logs.

# System Architecture

## Frontend (React)

## Backend (Express)

## Database (PostgreSQL)

## DevOps & CI/CD (Docker)

## Hosting & Development (To be conducted)

# UI/UX Design (Figma)

# Development Tools & Stack Summary

# Project Milestones

## Planning and Research

* Define the project requirements and create a detailed project plan.

## Design and Prototyping

* Develop the wireframes and prototypes for the UI.
* Start Implementation on Figma and gather feedback on the UI Design.

## Development

* Implement the core features of the software.
* Conduct regular testing and debugging to ensure functionality.

## Testing

* Allow selected group of users to try out the application.
* Collect feedback and make necessary improvements.

## Launch

* Launch the software to the public on specified platform.

## Final Documentation

* Finalise the Documentation.
* Implement ReadMe.