**COMP3059 Capstone Project I - Sprint 4**

**Part II - Technology Requirements and Learning Plan**

**Introduction**

This document outlines the technological requirements and learning plan for the worker scheduling application. It provides an overview of the technologies chosen for the project, their advantages and disadvantages, and the learning plans for team members.

**A) Technological Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Technology | Description | Pros | Cons |
| React.js | Frontend library for building dynamic web UIs. | - Component-based structure. - Excellent state management. | - Requires learning React concepts for new developers. |
| Node.js | JavaScript runtime for backend development. | - Single language for frontend and backend. - Scalable for real-time apps. | - Handles CPU-intensive tasks poorly. |
| Express.js | Minimalistic web framework for Node.js. | - Simplifies backend development. - Great for building REST APIs. | - Requires integration with other tools for full-stack functionality. |
| MongoDB | NoSQL database for flexible data storage. | - Schema flexibility for handling shift and schedule data. - Scales well for dynamic workloads. | - Querying is less straightforward compared to SQL databases. |
| Socket.IO | Real-time bidirectional communication framework. | - Ideal for chat functionality and live updates. - Lightweight and easy to integrate. | - Can have performance issues under heavy traffic. |
| Figma | UI/UX design tool for interactive prototypes. | - Easy collaboration with team. - Simple to share mockups. | - Some advanced features are premium. |
| Postman | API testing tool for backend endpoints. | - Simplifies API testing. - Great for ensuring robust backend services. | - Limited automation without premium subscription. |
| Docker | Containerization for consistent deployment. | - Simplifies environment management. - Works well across teams and machines. | - Steep learning curve for beginners. |
| Vercel | Cloud platform for deploying the application. | - Easy integration with Git. - Free tier available for small projects. | - Limited scalability and features in the free tier. |

**B) Learning Plan**

**Team members responsibilities and Skill Levels**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Responsibility | Skill Level (%) | Learning Plan |
| Meet | Backend Development and Integration | 80% | Learning Node.js and Express.js basics with Documentation. |
| Shyam | Testing and Deployment and real time communication | 85% | Learning Postman and Docker basics |
| Harin | Database Design and Backend Development | 85% | Master MongoDB schema design |
| Manvi | Frontend UI/UX Development | 80% | Practice and learned React.js fundamentals. |

**Team members responsibilities and Skill Levels**

|  |  |  |  |
| --- | --- | --- | --- |
| Learning Topic | Start Date | End Date | Resources |
| Node.js and Express.js Basics | September 6th | December 15th | Full Stack Development course and node.js documentation |
| React.js Development | October 2nd | December 15th | React Official Docs and Full Stack Development Courses |
| MongoDB Schema Design | September 6th | December 15th | MongoDB Website and SpringBoot/ MERN Course. |
| Socket.IO for Chat Features | December 15th | January 1st | Socket.IO Get Started. |
| Postman for API Testing | September 15th | December 15th | Spring Boot Course and Postman Tutorials and Documentation. |
| Docker for Deployment | September 15th | December 15th | Docker Documentation and YouTube and MERN Course. |