**Business Requirements Specification (BRS)**

**Task Management System (TMS) Frontend**

**Document Version**: 1.0  
**Date**: June 12, 2025

**1. Introduction**

**1.1 Purpose**

This Business Requirements Specification (BRS) document outlines the business objectives, functional requirements, and high-level specifications for the Task Management System (TMS) frontend hosted at <https://fk-tms-frontend.vercel.app/>. The TMS aims to provide an intuitive, scalable, and responsive web application to manage tasks efficiently for individual and team-based workflows.

**1.2 Scope**

The TMS frontend will enable users to create, view, update, and delete tasks, manage task statuses, and collaborate with team members. It will leverage Vercel’s Frontend Cloud for seamless deployment, scalability, and performance optimization. The application will focus on user experience, accessibility, and integration with backend APIs for data management.

**1.3 Definitions, Acronyms, and Abbreviations**

* **TMS**: Task Management System
* **Frontend Cloud**: Vercel’s platform for hosting and deploying frontend applications
* **API**: Application Programming Interface
* **UI/UX**: User Interface/User Experience
* **CRUD**: Create, Read, Update, Delete
* **SPA**: Single Page Application

**1.4 References**

* Vercel Documentation: [https://vercel.com/docs[](https://vercel.com/docs)](https://vercel.com/docs%5B%5D(https:/vercel.com/docs))
* Frontend Development Best Practices: [https://thenewstack.io](https://thenewstack.io/)

**2. Business Objectives**

The TMS frontend aims to achieve the following business goals:

1. **Streamline Task Management**: Provide an intuitive interface for users to manage tasks efficiently, reducing manual effort and improving productivity.
2. **Enhance Collaboration**: Enable team-based task assignment, tracking, and communication.
3. **Ensure Scalability**: Leverage Vercel’s Frontend Cloud for high availability, fast load times, and global accessibility.
4. **Improve User Experience**: Deliver a responsive, accessible, and visually appealing interface to maximize user adoption.
5. **Support Integration**: Seamlessly connect with backend APIs for data persistence and third-party integrations (e.g., calendar, notifications).

**3. Functional Requirements**

**3.1 User Management**

* **BR-001**: Users shall register and log in using email/password or third-party authentication (e.g., Google, GitHub).
* **BR-002**: Users shall have personalized dashboards displaying their tasks and team tasks.
* **BR-003**: Role-based access control (e.g., Admin, Team Member) shall restrict certain actions (e.g., task deletion, user management).

**3.2 Task Management**

* **BR-004**: Users shall perform CRUD operations on tasks (create, read, update, delete).
* **BR-005**: Tasks shall include attributes such as title, description, due date, priority, status (e.g., To-Do, In Progress, Done), and assignees.
* **BR-006**: Users shall filter and sort tasks by status, priority, or due date.
* **BR-007**: Users shall assign tasks to themselves or team members.

**3.3 Collaboration Features**

* **BR-008**: Users shall comment on tasks to facilitate team communication.
* **BR-009**: The system shall notify users of task updates (e.g., assignment, status change) via email or in-app notifications.
* **BR-010**: Users shall view a shared team calendar displaying task deadlines.

**3.4 UI/UX Requirements**

* **BR-011**: The frontend shall be a responsive SPA built with a modern JavaScript framework (e.g., React, Next.js).
* **BR-012**: The interface shall support light and dark modes for user preference.
* **BR-013**: The application shall comply with WCAG 2.1 accessibility standards for inclusivity.

**3.5 Deployment and Performance**

* **BR-014**: The application shall be hosted on Vercel’s Frontend Cloud for automatic scaling and global CDN distribution.
* **BR-015**: Deployments shall support preview environments for testing changes before production.
* **BR-016**: The frontend shall integrate with backend APIs using REST or GraphQL for data operations.

**4. Non-Functional Requirements**

**4.1 Performance**

* **BR-017**: Page load times shall not exceed 2 seconds under normal conditions.
* **BR-018**: The application shall handle up to 1,000 concurrent users without performance degradation.

**4.2 Security**

* **BR-019**: User data shall be protected with HTTPS and secure authentication mechanisms.
* **BR-020**: The frontend shall prevent common vulnerabilities (e.g., XSS, CSRF) through best practices.

**4.3 Scalability**

* **BR-021**: The system shall scale automatically with Vercel’s serverless architecture to handle traffic spikes.

**4.4 Compatibility**

* **BR-022**: The frontend shall support the latest versions of Chrome, Firefox, Safari, and Edge browsers.
* **BR-023**: The application shall be responsive across desktop, tablet, and mobile devices.

**5. Assumptions and Constraints**

**5.1 Assumptions**

* The backend APIs required for task management and user authentication are available and documented.
* Users have JavaScript enabled in their browsers to access the application.
* Vercel’s Frontend Cloud will provide the necessary infrastructure for hosting and scaling.

**5.2 Constraints**

* The application must be developed using a JavaScript-based framework compatible with Vercel (e.g., React, Next.js).
* No local file I/O or direct database connections are allowed in the frontend; all data operations must go through APIs.
* The project timeline and budget are not specified in this BRS and will be determined in subsequent planning phases.

**6. Stakeholders**

* **Project Sponsor**: Funnels Kingdom
* **Development Team**: Frontend developers, UI/UX designers, QA engineers
* **End Users**: Individuals and teams using the TMS for task management
* **Vercel Support**: Provides hosting and deployment infrastructure

**7. Risks and Mitigation**

* **Risk 1**: Backend API delays may impact frontend development.
  + **Mitigation**: Develop mock APIs for frontend testing during backend development.
* **Risk 2**: Users without JavaScript enabled cannot access the application.
  + **Mitigation**: Display a fallback message prompting users to enable JavaScript.
* **Risk 3**: Performance issues under high traffic.
  + **Mitigation**: Leverage Vercel’s edge network and caching for optimized performance.

**8. Acceptance Criteria**

* The TMS frontend shall allow users to perform all CRUD operations on tasks.
* The application shall load successfully on supported browsers with JavaScript enabled.
* The frontend shall integrate seamlessly with backend APIs for data retrieval and updates.
* Deployments on Vercel shall complete without errors, with preview and production environments accessible.
* The UI shall be responsive and meet WCAG 2.1 accessibility standards.

**9. Future Considerations**

* Integration with third-party tools (e.g., Slack, Google Calendar) for enhanced collaboration.
* Addition of offline support using Progressive Web App (PWA) features.
* Implementation of advanced analytics for task completion trends and user productivity.

**10. Approval**

**Prepared by**: Sarim Nadeem (SQA & PM)  
**Reviewed by**: Sarim Nadeem  
**Date**: June 12, 2025