**Business Requirements Document (BRD) - Project Management Web Application**

**1. Project Overview**

The Project Management Web Application is a comprehensive project management tool inspired by JIRA, ASANA, TRELLO designed to help teams efficiently manage their projects, track tasks, and enhance collaboration.

**### Objective**

The primary objective of this project is to develop a feature-rich project management web application that enables teams to plan, execute, and monitor their projects seamlessly. The application will provide an intuitive user interface and robust functionality to streamline project workflows.

**### Stakeholders**

The key stakeholders for this project include:

- Product Owner

- Project Managers

- Development Teams

- Quality Assurance Team

- End Users (Internal Teams, Project Contributors)

**2. Project Goals and Objectives**

**### Goals**

1. Develop a feature-rich project management tool with task tracking, issue management, and team collaboration capabilities.

2. Enable real-time updates and seamless user interactions to improve project workflows.

3. Provide high-quality user interfaces that enhance user experience and ensure smooth navigation across features.

**### Objectives**

1. Centralize project and task management within a single platform.

2. Enhance team productivity by providing tools for efficient communication, task delegation, and progress monitoring.

3. Maintain high performance and reliability, supporting concurrent users and real-time data updates.

**3. Functional Requirements**

**### 3.1 User Management and Access Control**

- User Registration and Login: Users should be able to sign up and log in securely.

- Role-based Access: Implement role-based access (e.g., Admin, Project Manager, Team Member).

- Role Assignment: Assign users to specific roles and project teams with appropriate access controls.

**### 3.2 Project Management**

- Project Creation and Deletion: Users with the necessary permissions can create, edit, or delete projects.

- Project Overview Dashboard: Provide a project overview dashboard that shows active projects, timelines, assigned members, and current progress.

- Task and Subtask Management: Create, assign, update, and delete tasks, including the ability to organize tasks hierarchically with subtasks.

- Status Tracking: Assign status indicators to tasks (e.g., To Do, In Progress, Done, Blocked) and allow users to update the status as tasks progress.

**### 3.3 Issue and Bug Tracking**

- Issue Creation and Assignment: Users can log issues, assign them to team members, and set due dates.

- Priority Levels: Allow users to set issue priority levels (e.g., Low, Medium, High, Critical).

- Comments and Activity Log: Enable comments and activity logging for issues to improve communication.

- Resolution Tracking: Track issue status from creation to resolution.

**### 3.4 Reporting and Analytics**

- Progress Tracking: Track project and task progress visually with metrics like % completed and time remaining.

- Productivity Reports: Generate reports on team productivity, issue resolution times, and project timelines.

- Customizable Dashboards: Allow users to personalize dashboards with specific metrics and project views.

**### 3.5 Notifications and Alerts**

- Real-time Notifications: Notify users about task assignments, project updates, and deadline reminders.

- Email and Push Notifications: Send notifications via email or push to keep users updated on important actions.

**### 3.6 Integrations and API**

- Third-Party Integrations: Integrate with third-party tools (e.g., Slack, GitHub, Google Calendar).

- REST API: Provide an API for external systems to access or update data (e.g., project details, task status).

**4. Non-Functional Requirements**

**### 4.1 Usability**

- Intuitive User Interface: The application should have a user-friendly and visually appealing interface, minimizing the learning curve for new users.

- Accessibility: Ensure the application is accessible to users with disabilities (e.g., WCAG compliance).

**### 4.2 Performance**

- Response Time: All interactions should load within 2 seconds under normal load conditions.

- Scalability: Support concurrent access for large teams (up to 500 users simultaneously).

- Load Handling: Maintain stable performance under increased load (up to 1,000 tasks per project).

**### 4.3 Security**

- Data Encryption: Encrypt sensitive data in transit (e.g., using HTTPS) and at rest (e.g., database encryption).

- Role-Based Access Control: Enforce strict access control for user roles.

- Audit Logging: Track user actions for security and compliance.

**### 4.4 Reliability and Availability**

- High Availability: Ensure 99.9% uptime for critical features.

- Backup and Recovery: Daily data backups and efficient recovery plans in case of data loss.

**### 4.5 Maintainability**

- Modular Codebase: Develop a modular codebase for ease of maintenance and feature expansion.

- Documentation: Provide detailed technical documentation for each module.

**### 4.6 Compatibility**

- Browser Compatibility: Compatible with modern web browsers (Chrome, Firefox, Edge, Safari).

- Device Compatibility: Optimize for both desktop and mobile views to ensure functionality on all devices.

**5. Assumptions and Constraints**

- The system should support multiple teams and projects simultaneously.

- The system should be built with a flexible architecture for future feature additions.

- Development will follow Agile methodologies, with iterative releases for functionality validation.

**6. Success Metrics**

- User Satisfaction: Measure user feedback for usability and functionality.

- System Performance: Track response times and downtime events.

- Feature Adoption: Monitor the use of key features (e.g., task creation, issue tracking).

- Error Rates: Minimize bug reports and system crashes to less than 1% of interactions.

**7. Risks and Mitigations**

**### Risks and Mitigations**

- Risk: High user traffic may overload the system.

  Mitigation: Implement load balancing and caching to manage high traffic.

- Risk: Security breaches due to unauthorized access.

  Mitigation: Enforce multi-factor authentication and regular security audits.