

TestSprite

AI-Powered End-to-End Testing Platform

Product Specification Document (With System Architecture)

Prepared by: Product Management Team

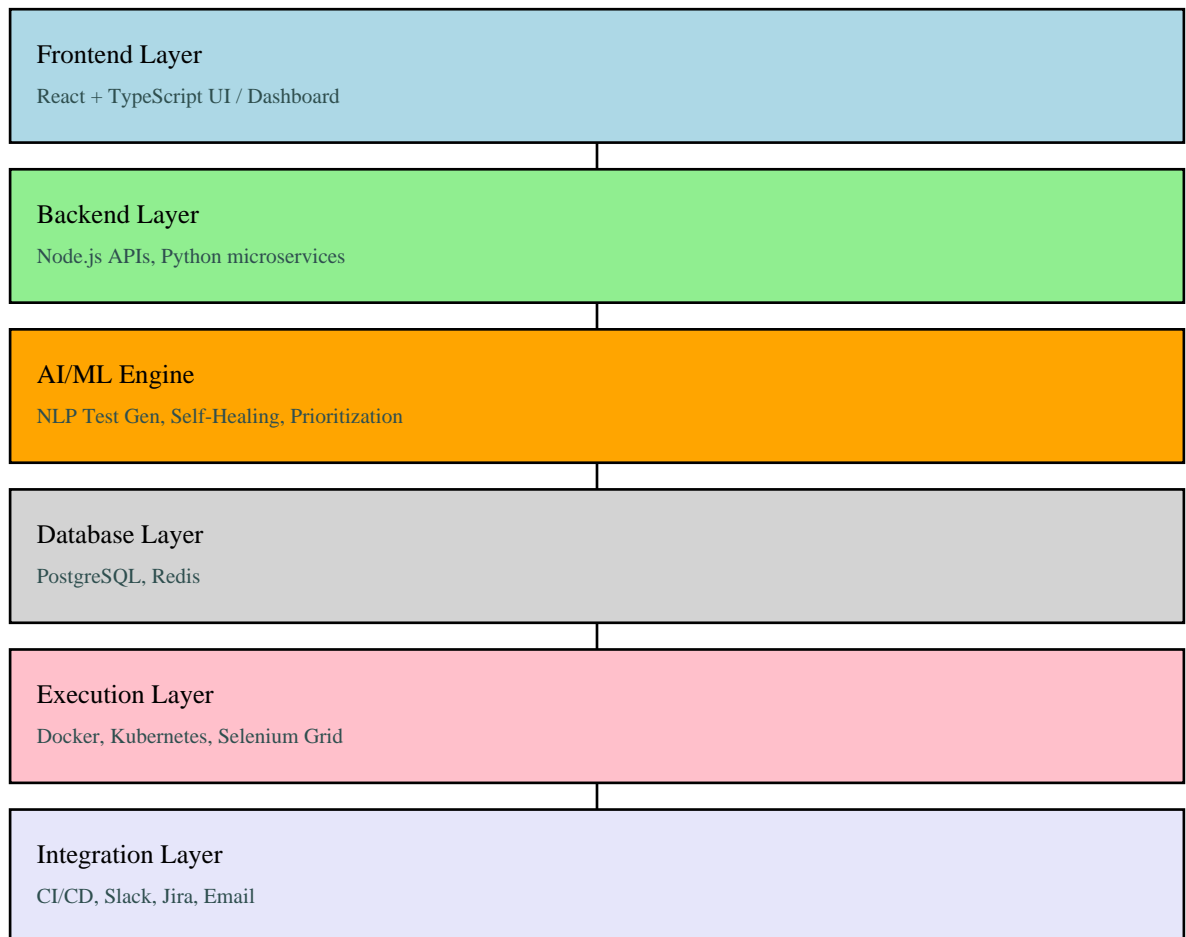
Version: 1.1

Date: October 2025

8. System Architecture Overview

The TestSprite system follows a modular, microservice-based architecture that integrates AI-powered modules with CI/CD pipelines and scalable test execution environments.

Below is a visual representation and description of the architecture layers.



Architecture Layers:

- Frontend Layer:** Built with React and TypeScript for an interactive dashboard, no-code builder, and AI assistant interface.
- Backend Layer:** Node.js orchestrates API requests and Python microservices handle AI tasks and test orchestration.
- AI/ML Engine:** Uses NLP and ML models for automated test case generation, self-healing tests, and prioritization.

4. **Database Layer:** Stores project data, test runs, analytics, and session data using PostgreSQL and Redis.

5. **Execution Layer:** Containerized test environments managed via Docker and Kubernetes with Selenium Grid, Playwright, and Appium.

6. **Integration Layer:** Enables CI/CD tool integration with Jenkins, GitHub Actions, and notifications via Slack and Jira.

9. Data Flow and Process

1. User logs into TestSprite and connects a project repository. 2. AI engine scans code changes and generates relevant test cases. 3. Tests are executed in containerized environments with dynamic resource allocation. 4. Results and logs are sent back to the dashboard. 5. AI learns from test outcomes to enhance future predictions and script healing. 6. Reports and insights are distributed through CI/CD tools and collaboration channels.