

A web application consists of a frontend built using React and a backend developed with Node.js and Express. The frontend interacts with the backend via a REST API, which handles HTTP requests for user authentication, data retrieval, and business logic execution. The backend connects to a MongoDB database, which stores user information, transaction records, and system logs.

Users authenticate using JWT-based authentication, ensuring secure access control. Passwords are hashed using bcrypt before storage. Role-based access control (RBAC) is implemented, allowing different user roles such as Admin, Editor, and Viewer, each with specific permissions.

The application is hosted on AWS, utilizing S3 for file storage and EC2 for server deployment. A load balancer distributes incoming traffic to multiple EC2 instances, ensuring high availability. API Gateway is used to manage API requests, and Lambda functions are deployed for serverless execution of background tasks.

For performance optimization, a Redis cache is integrated to reduce database queries by caching frequently accessed data. WebSocket connections are used for real-time updates, enabling features like live notifications and chat functionality.

The system is monitored using Prometheus for collecting metrics and Grafana for visualization. Logging is managed using the ELK stack (Elasticsearch, Logstash, and Kibana), allowing efficient debugging and analysis of system behavior.

CI/CD pipelines are implemented with GitHub Actions and Docker, enabling automated testing and deployment. The backend services are containerized using Docker and orchestrated with Kubernetes, ensuring scalability and fault tolerance. The application follows a microservices architecture, with independent services handling authentication, payment processing, and analytics.