Scalability & Availability Considerations

Handling Increased Users or Data

The *MyDormStore* system is designed with scalability in mind. The backend is built with Express.js and PostgreSQL, and all database queries are optimized to handle large datasets efficiently—such as a growing catalog of dorm products and user orders. Pagination is implemented for product and order lists to avoid overloading the UI or backend with large payloads.

The frontend, built in React, relies on modular components that render dynamically based on API responses, making it responsive even as user volume grows. We've ensured that features like the admin dashboard, order tracking, and checkout remain performant as the number of users and data entries increase.

Load Balancing & Caching

While we currently rely on single-instance deployments, the app architecture supports future horizontal scaling through load balancing (e.g., using Nginx or cloud-managed load balancers). This would allow us to distribute traffic across multiple backend instances during high-traffic periods such as move-in season.

Caching mechanisms (such as HTTP caching or local storage for dorm selection and user profile data) reduce repeated calls to the database for frequently accessed information, minimizing server load and response times.

Deployment Strategy & Uptime

MyDormStore is deployed using a cloud-based platform (e.g., Railway), which handles continuous deployment from our GitHub repository. The CI/CD pipeline ensures that only tested and stable code is deployed, reducing downtime from broken features.

We also plan to implement the following strategies to improve uptime and reliability:

- PR Freeze Policy: No merges allowed 24 hours before sprint deadlines to allow for full testing and prevent last-minute errors.

- Rollback Support: In the event of a failed deployment, previous working versions can be redeployed quickly.

- Monitoring (Future Work): Integrating tools like UptimeRobot or Render monitoring to detect crashes or downtime in real time.

With these strategies in place, *MyDormStore* is capable of scaling smoothly with growing user demand while maintaining high availability.