**Improvements for the FrontEnd Vue.js app**

1. Minify CSS and JavaScript
2. Use dynamic imports to reduce bundle size

const TableView = () => import('./components/TableView.vue');

1. Store sensitive information like API keys in Azure or the GitHub repository's secret manager, never in the source code or .env files.
2. Add a CSP header in Azure configuration to restrict what external resources the app can load.
3. Display meaningful error messages when API requests fail.
4. For large datasets, implement pagination on the table and the details view.
5. Integrate tools like Sentry or Azure Application Insights to capture frontend errors.
6. Write end-to-end tests with tools like [Playwright](https://playwright.dev/).

**Improvements for the Web API app:**

1. Replace the basic API key authentication with more robust options like JWT (JSON Web Tokens), OAuth2, or Azure AD.
2. Use role-based or claim-based authorization to control access to endpoints based on user roles.
3. Ensure all input data is validated to prevent invalid data and malicious input.
4. Handle error and exceptions properly
5. Add monitoring tools like Application Insight to monitor the service
6. In production, avoid exposing stack traces or internal details in error messages.
7. Cache API responses for frequently requested data to improve performance and reduce server load.(Redis , Response Caching)
8. Store sensitive data (e.g., API keys, connection strings) securely using environment variables or Azure Key Vault
9. Use a Database for Real Data
10. Configure CORS to allow only trusted origins instead of using AllowAnyOrigin.
11. Configure Azure App Service with scaling rules to handle increased traffic
12. Use API versioning to ensure backward compatibility when making changes.