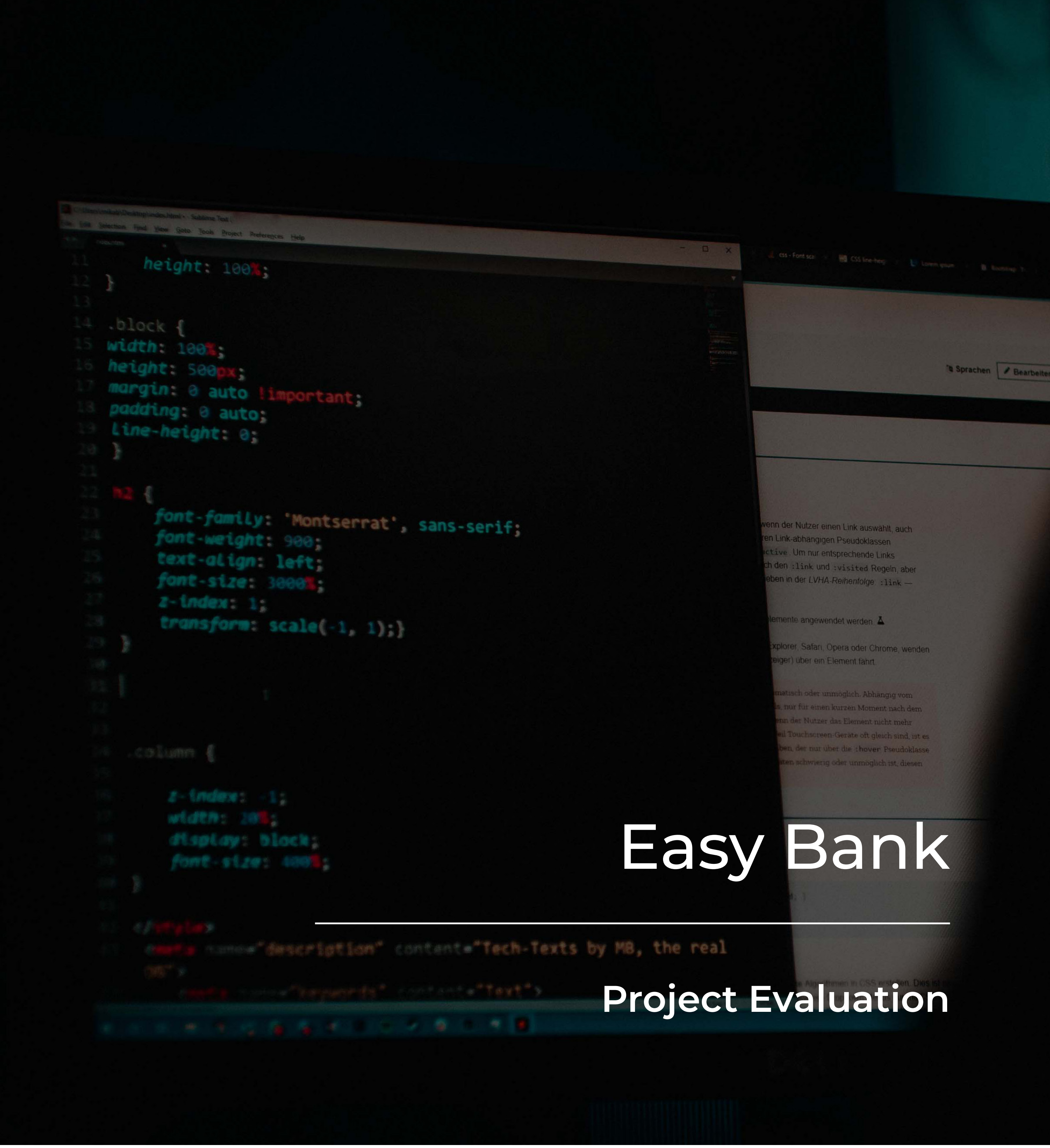

Easy Bank

Project Evaluation

Kushan Wijesinghe | iAMDT.3073

Development Process Analysis

The development process for the Easy Bank app underwent a systematic approach following a phased timeline. The planning and design phase were comprehensive, aiding in clear requirement gathering and a robust architecture. However, the development phase encountered some delays due to unforeseen technical challenges, which impacted meeting deadlines.

Outcome and Milestones

The app's outcome was a user-friendly, feature-rich mobile banking platform. Despite encountering delays, the final product met most milestone objectives, although some features needed post-launch enhancements. The interface design was intuitive, but some refinements were required for a more seamless user experience.

Strategies for Future Improvement

To enhance future development projects, a more thorough risk assessment during the planning phase could mitigate unexpected delays. Agile methodologies could be implemented for better adaptability to evolving requirements, ensuring prompt adjustments during development.

Code and Interface Development Review

Code development followed industry best practices, ensuring scalability and maintainability. However, meeting milestones and deadlines posed challenges due to unforeseen complexities in integrating certain functionalities. Interface development aligned with design expectations, but user testing revealed areas for improvement in usability and responsiveness.

App Development Framework Analysis

The chosen React Native framework facilitated cross-platform development, enabling faster deployment across multiple devices. Its component-based structure streamlined development, but certain platform-specific features required additional customization, causing delays.

Evaluation of Alternative Development Framework

Comparatively, using a native development framework (e.g., Swift for iOS, Kotlin for Android) might have offered deeper access to platform-specific features, potentially reducing post-launch refinements. However, it could have prolonged development time significantly due to separate codebases for each platform, impacting time-to-market and overall project cost.

Impact on Future Projects

In future projects, balancing the advantages of a cross-platform framework like React Native with the specific needs of the app will be crucial. A hybrid approach, leveraging React Native for broad functionality and native development for critical features, could be explored to optimize both development time and app performance.