




STRUCTURED EXAM OVERVIEW

ST10153600- DAVID SADIKI
ST1005677-MBALI BANDA
ST10175937-OLWETHU NTOBELA
ST10192784-MOSEBYADI LEGODI



Structured Exams App Overview

The Structured Exams app is designed to help students manage their exam schedules efficiently, providing features for task tracking, countdowns, and progress monitoring. Key design considerations include a user-friendly interface, pastel color scheme, and accessibility features to enhance usability. The app is hosted on GitHub, utilizing version control for collaborative development. Automation is implemented using GitHub Actions to streamline testing and deployment processes.

UI/UX User interface- Olwethu

Utilization of AI Tools for XML Generation

AI tools have been employed to enhance the XML generation process by providing intelligent suggestions and automating repetitive tasks. These tools analyze existing code and design patterns, enabling developers to create XML layouts efficiently, ensuring adherence to best practices while minimizing errors. This integration of AI significantly streamlines the UI development process, improving productivity and fostering innovation in application design.

In the XML layout design for the Structured Exams application, AI was utilized in several key areas to enhance the development process:

1. **Automated Layout Suggestions:** AI tools provided intelligent suggestions for organizing UI elements, such as aligning buttons and text fields according to design principles. This ensured a visually appealing and user-friendly interface.
2. **Error Detection and Correction:** AI tools analysed the XML code for common errors, such as missing attributes or incorrectly nested elements. This helped developers identify and rectify issues before testing the app, leading to a smoother development process.

Backend Development- David

In the development of the **Structured Exams** application, I contributed to establishing a robust backend infrastructure to support essential features such as task management and countdown displays. The backend development is centered on efficient data handling, secure user authentication, and dynamic task retrieval.

Backend Overview: A critical aspect of this project is implementing secure user authentication. This process involves user registration and login, which includes hashing passwords to safeguard sensitive information and issuing tokens for authenticated sessions. By leveraging secure authentication methods, we enhance the overall security of user data, protecting against unauthorized access. Additionally, the backend is designed to facilitate the retrieval of tasks and countdowns in real-time, ensuring users have immediate access to their academic

schedules. This comprehensive approach not only improves the user experience but also establishes a reliable and secure foundation for the Structured Exams application.

Scrum Master- Mbali

In my part, I compared three mobile applications—My Study Life, Exam Countdown, and Countdown Widget—focusing on their features, target users, and functionalities. To assist with this comparison, I utilized AI technology for generating insights and organizing the comparison in a clear and structured way.

AI was specifically employed in the following areas:

1. **Data Gathering and Structuring:** AI was used to collect and compile relevant information about the three apps. It synthesized the key features of each app and organized the data into clear sections for easy comparison. By using AI, I was able to gather details from multiple sources and streamline the key points.
2. **Analysis and Comparison:** AI helped in analysing the different aspects of the apps, identifying commonalities, and highlighting differences in their target audiences, primary functionalities, and use cases. It also assisted in generating clear, concise paragraphs that summarize these differences logically.

API- Mosebyadi

AI assistance was utilized for structuring the setup process, including the initialization of Firebase Admin SDK and the integration of Firebase Authentication. The AI provided guidance on the necessary Gradle dependencies, code structure for user registration and login routes, as well as ensuring secure communication with Firebase. Additionally, AI helped streamline the task management feature with Firebase Realtime Database, suggesting code for reading, writing, and querying tasks. This assistance accelerated development and ensured adherence to best practices for secure authentication and efficient data management.

1. **Gradle Dependencies:** AI suggested the necessary Gradle dependencies for Firebase integration.
2. **Code Structure Guidance:** AI offered recommendations for the code structure for user registration and login routes.