**TeachHub Project - Implementation Plan**

**1. Introduction**

TeachHub is an educational testing platform designed to help teachers create exams and track student performance. The project relies on **Firebase** as the backend, with a user interface built using **React.js, Bootstrap, and Tailwind CSS**, ensuring a smooth and user-friendly experience.

**2. Project Planning & Management**

**Project Goals**

* Develop an easy-to-use testing platform for teachers and students.
* Provide individual student accounts for tracking progress and results.
* Enhance security to prevent cheating and question extraction.
* Improve user experience with a fast and responsive interface.

**Task Management**

* Divide the project into phases using **Agile/Scrum** methodology.
* Use **Notion** or **Trello** to track progress.
* Set deadlines for each development phase.

**3. Literature Review**

**Analysis of Similar Platforms**

* Review **Google Classroom, Kahoot, Quizizz** to identify strengths and weaknesses.
* Analyze **anti-cheating** methods in online exams.
* Evaluate best practices for improving **User Experience (UX/UI)**.

**Enhancing User Experience**

* Simplify the exam creation process for teachers.
* Improve test navigation and result reporting for students.
* Support **Dark Mode** for better accessibility.

**4. Requirements Gathering**

**Target Users**

* **Teachers**: Create tests, track student performance, and analyze results.
* **Students**: Take tests, review answers, and improve their skills.
* **Administrators**: Manage users, generate reports, and monitor platform performance.

**Software Requirements Specification (SRS)**

**Core Features**

* Create, edit, and delete tests.
* Set test time limits and deadlines.
* Support multiple question types (MCQs, essays, true/false).
* Provide performance statistics for students.

**Security Requirements**

* Protect data using **Firebase Authentication**.
* Secure communication with **HTTPS and JWT Tokens**.
* Prevent cheating by disabling code inspection and hiding test questions in the source code.

**Performance Requirements**

* Support a high number of simultaneous users.
* Ensure fast page loading with **Firebase Hosting**.
* Optimize database queries using **Firestore Indexing**.

**Methods for Requirements Gathering**

* Conduct surveys and interviews with teachers to understand their needs.
* Collect student feedback on existing challenges in online testing.
* Analyze usage data to identify areas for improvement.

**5. System Analysis**

**System Architecture Diagrams**

* **Use Case Diagram** illustrating user roles in the system.
* **Entity Relationship Diagram (ERD)** defining the database structure.
* **Flowchart** showing the system’s workflow.
* **Sequence Diagram** explaining the step-by-step execution of key operations.

**Technologies Used**

* **Frontend**: React.js, Bootstrap, Tailwind CSS.
* **Backend**: Firebase Firestore, Firebase Authentication, Firebase Functions.
* **Database**: Firestore.
* **Authentication System**: Firebase Authentication.
* **Data Security**: Encryption using **JWT & HTTPS**.

**6. Next Steps**

* Develop a prototype to test core functionalities.
* Conduct trials with a selected group of teachers and students.
* Improve security and performance based on user feedback.
* Implement additional features such as **real-time notifications, Google login, and LMS integration**.

**This document has been prepared to outline the implementation phases of TeachHub and ensure its success in achieving the intended goals.**