**App Documentation Profile**

This document outlines the architecture, features, and functionality of the application. It is intended to provide a clear and comprehensive description of the software as it exists at the time of this document's creation and may be referenced in legal agreements.

**Phase 1 Plan**

This phase focuses on core user management and the basic LMS structure.

1. **Login and Sign-Up:**
   * The application provides functionality for users to create new accounts and securely log in.
   * Functionality includes:
     + Input validation to ensure data integrity.
     + Password hashing using industry-standard algorithms (e.g., bcrypt, Argon2) to protect user credentials.
     + Session management to maintain user authentication state.
2. **Database and Schema:**
   * A database will be designed to store user information, course data, and other relevant data, as defined in the schema.
   * The schema will define the structure of the database tables, including:
     + Data types for each field.
     + Relationships between tables.
     + Constraints to ensure data integrity (e.g., primary keys, foreign keys, unique constraints).
3. **Test Cases:**
   * Comprehensive test cases will be developed to ensure the reliability and robustness of the application.
   * This includes:
     + Unit tests to verify the functionality of individual components.
     + Integration tests to verify the interaction between different components.
     + End-to-end tests to verify the overall system functionality.
     + Specific test cases will be created for login, sign-up, database operations, and schema validation.
4. **LMS System Framework:**
   * The basic framework for the Learning Management System (LMS) will be established.
   * This framework includes:
     + Definition of user roles (student, teacher, admin) with associated privileges.
     + Core functionalities for each user role, as described below.

**User Flow**

The primary user flow for a student is as follows:

1. **Sign Up/Login:**
   * New users shall create an account by providing required information.
   * Existing users shall log in using valid credentials.
   * Successful authentication grants access to the application's features, as defined by the user's role.
2. **Dashboard:**
   * Upon successful login, users are directed to their personalized dashboard.
   * The dashboard provides an overview of the user's profile and available courses.
3. **Payment:**
   * (Functionality to be implemented).
   * This will enable users to make payments for services within the application, such as course enrollment.
   * Specific payment methods and security measures will be detailed in subsequent documentation.

**User Roles and Functionality**

The application supports the following user roles, each with specific functionalities:

**1. Student**

* **Dashboard:**
  + Provides an overview of the student's profile and available courses.
  + **Student Profile Data:**
    - The following student information is stored in the database: username, id (self-generated unique identifier), phone number, address, password (stored as a cryptographic hash), sessionTokenAt (timestamp of session creation), and JWT (JSON Web Token for authentication).
    - The dashboard displays:
      * ID (Self-generated)
      * USERNAME
      * Address
      * Phone Number
  + **Courses:**
    - Students can access and manage courses.
    - The application provides two categories of courses:
      * **Subscribed Courses:** A list of courses in which the student is currently enrolled.
      * **Saved Courses:** A list of courses the student has saved for future enrollment.
    - Courses are displayed, taking into account factors such as:
      * Date of addition (newly added courses).
      * Relevance to the student's interests (where applicable).
      * Other sorting criteria as determined by the application's logic.
  + **Videos:**
    - Students can access video content within courses.
    - Functionality includes:
      * Browsing and scrolling through available videos.
      * Selecting specific videos for viewing.
      * Playing selected videos using an integrated media player.
      * The specific media player and its features are subject to change and will be documented in subsequent revisions.
  + **Question List:**
    - Students can manage their questions.
    - The application supports the following categories of questions (represented by an enumerated type): questionAskedByStudent, Liked, and Saved.
    - Each question includes:
      * The text of the question.
      * A video answer (to be implemented in Phase 2, potentially using AI-driven functionality).
  + **Ask a Question:**
    - Students can ask questions and direct them to:
      * The community (a public forum for student interaction).
      * A specific teacher (for direct communication).
      * The technical support team (for technical assistance).

**2. Teacher**

* **Dashboard:**
  + Provides an overview of the teacher's profile and their courses.
  + Functionalities include:
    - Managing courses (e.g., creating, editing, deleting).
    - Uploading content (e.g., videos, documents).
    - Interacting with students (e.g., answering questions, providing feedback).
  + The dashboard includes sections for:
    - BIO (biographical information).
    - Videos (uploaded by the teacher).
    - Courses (managed by the teacher).
    - PDF Text Workbook Courses (materials provided by the teacher).
    - The specific features and layout of the teacher's dashboard may be further defined in subsequent documentation.

**3. Admin**

* The functionalities and permissions for the Admin role will be defined in Phase 2.
* It is anticipated that the Admin role will have broad privileges, including user management, system configuration, and content management.

**Legal Considerations**

This document is intended to provide a functional description of the application. It is not intended to be a comprehensive legal agreement. The following points should be considered when incorporating this document into a legal contract:

* **Scope:** This document describes the application's functionality as of the date of its creation. Future changes and updates to the application may not be reflected in this document.
* **Acceptance:** The parties to the legal agreement should agree that this document accurately reflects the agreed-upon functionality of the application.
* **Modifications:** Any modifications or deviations from the functionality described in this document should be documented and agreed upon in writing by all parties.
* **Warranties:** Any warranties or guarantees regarding the application's performance, reliability, or fitness for a particular purpose should be explicitly stated in the legal agreement.
* **Intellectual Property:** The legal agreement should clearly define ownership of intellectual property related to the application, including software, documentation, and any related materials.