**Multi-Page Flet Application**

**A Technical Overview**

**Slide 1: Title Slide**

* **Title:** Multi-Page Flet Application
* **Subtitle:** User Authentication, Dashboard, and Note Management System
* **Presented by:** [Your Name]
* **Date:** [Presentation Date]

**Slide 2: Introduction**

* **Purpose of the Application**:
  + Provide a structured multi-page application for user management and note-taking.
  + Build a cohesive system that includes authentication, data management, and user-friendly navigation.
* **Technologies Used**:
  + **Flet Framework:** For front-end development and UI.
  + **Python & SQLite:** For backend and data management.
  + **dotenv:** For configuration management and security.

**Slide 3: Architecture Overview**

* **Key Components**:
  + **Router**: Manages page navigation.
  + **Pages Directory**: Each page (Login, Signup, Dashboard, Posting) is represented by a Python class.
  + **Utilities Directory**: Houses helper functions for database interaction, validation, and styling.
* **Routing**:
  + Each route corresponds to a specific page (e.g., /dashboard or /posting), making the app easily navigable.

**Slide 4: Router Class**

* **Responsibilities**:
  + Initializes routes and assigns each route to a corresponding page view.
  + Manages the page transition by clearing previous views and loading new ones.
* **Key Methods**:
  + route\_change(): Detects route changes and loads the appropriate page view.
  + app\_router: A dictionary that maps each route to its respective page view.

**Slide 5: Pages Overview**

1. **LoginPage**:
   * **Purpose**: User authentication.
   * **Features**: Takes in credentials, verifies the user, and initiates a session.
2. **SignupPage**:
   * **Purpose**: User registration.
   * **Features**: Collects user information and saves it to the database.
3. **DashboardPage**:
   * **Purpose**: Main navigation page.
   * **Features**: Channel and bot token settings, sidebar for menu options, and basic user profile display.
4. **PostPage**:
   * **Purpose**: Note management.
   * **Features**: Allows users to add, edit, prioritize, and delete notes.

**Slide 6: Login and Signup Pages**

* **LoginPage**:
  + Contains input fields for username and password.
  + If valid, redirects the user to the dashboard.
* **SignupPage**:
  + Collects essential user details for registration.
  + Validates input and stores data securely in the database.
* **Session Management**:
  + Uses page.session to track user authentication status across routes.

**Slide 7: Dashboard Page**

* **Purpose**: Main control panel for the user after logging in.
* **Key Features**:
  + **Token and Channel Link Input**: Editable fields to configure the bot token and channel link, stored in the .env file.
  + **Sidebar Navigation**: Easy access to dashboard options and posting page.
  + **Profile and Notifications**: Basic user profile info and notification icon.

**Slide 8: PostPage - Note Management**

* **Purpose**: Centralized note-taking system.
* **Key Components**:
  + **Note Input Field**: Multi-line field for users to enter a note.
  + **Priority Selection**: Dropdown to assign priority (Low, Medium, High).
  + **Notes Display**: Lists saved notes with priority and date sorting options.
  + **Database Integration**:
    - Database.create\_note(): Adds new notes to the database.
    - Database.get\_user\_notes\_sorted(): Retrieves sorted notes based on user selection.
    - Database.delete\_note(): Allows for note deletion.
* **Dynamic UI Updates**:
  + Automatically refreshes when a new note is added or an old one is deleted.

**Slide 9: Database and Utilities**

* **Database**:
  + Stores user information, tokens, and notes.
  + **CRUD Operations**: Provides functions to create, read, update, and delete notes and user data.
* **Validation Utility**:
  + Validates data entries across pages, especially for required fields and format compliance.
* **Style Utility**:
  + Centralizes styles for reuse across components, enabling consistent color schemes, fonts, and element layouts.

**Slide 10: Configuration Management**

* **dotenv for Secure Configurations**:
  + Loads TOKEN\_BOT and CHANNEL\_LINK variables securely from a .env file.
  + Allows modifications from the dashboard and saves updated values to the .env file.
* **Session Variables**:
  + Tracks logged-in user data to provide a personalized experience.

**Slide 11: Demo**

* **Step-by-Step Walkthrough**:
  + Demonstrate the **Login** and **Signup** process.
  + Showcase the **Dashboard** with configuration settings.
  + Add and prioritize notes on the **PostPage** and view the saved notes.

**Slide 12: Error Handling and Logging**

* **Error Handling**:
  + Input validation is managed through the Validation utility.
  + Route management ensures a “404 Page Not Found” message for incorrect URLs.
* **Logging**:
  + Print statements are used for debugging configuration loading and session data.
  + Suggested Improvement: Use logging libraries for enhanced error tracking.

**Slide 13: Future Improvements**

* **Feature Additions**:
  + Enhance user profile management with settings and profile photo uploads.
  + Add functionality for editing notes.
* **UI Enhancements**:
  + Implement more advanced animations and improved styling.
* **Refined Logging**:
  + Replace debugging statements with structured logging.

**Slide 14: Q&A**

* **Questions?**
  + Invite the audience to ask questions or request further clarification on specific sections.

**Slide 15: Thank You**

* **Thank You for Your Attention**
  + Contact Information (e.g., email or LinkedIn for follow-up questions).