INSTALLATION GUIDE FOR THE DEVELOPMENT ENVIRONMENT AND APPLICATION USER MANUAL

OVERVIEW

01

INSTALLATION GUIDE

02

USER MANUAL

Q/A

Step 1: Install Flutter

1. Download Flutter SDK:
   - ○ Go to the official Flutter website and download the Flutter SDK for your operating system.
   - ○ Extract the downloaded ZIP file to a suitable location on your system (e.g., C:\src\flutter on Windows, /Users/yourname/flutter on macOS, or ~/development/flutter on Linux).

Set Up Environment Variable:

- Windows:
  - ○ Search for "Environment Variables" in the Start Menu.
  - ○ Click on "Edit the system environment variables."
  - ○ In the "System Properties" window, click "Environment Variables."
  - ○ Under "System Variables," select the Path variable and click "Edit."
  - ○ Click "New" and add the path to the Flutter SDK's bin directory (e.g., C:\src\flutter\bin).

- Open a new terminal or command prompt.
- Run the following command to verify the installation:

```
flutter doctor
```

- Flutter will check your environment and display a report. Ensure that all the checks are marked with a green checkmark. If there are any issues, follow the instructions provided to resolve them.

Step 2: Install Visual Studio Code

1. Download and Install VS Code:
    - Go to the official VS Code website.
    - Download the installer for your operating system and install VS Code.

Step 3: Install Flutter and Dart Plugins in VS Code

1. Open VS Code.
2. Install Flutter and Dart Extensions:
    - Click on the Extensions icon in the sidebar or press Ctrl+Shift+X (Cmd+Shift+X on macOS).
    - In the search bar, type Flutter and select the "Flutter" extension by "Dart-Code".
    - Click "Install" to install the Flutter extension. This will automatically install the Dart extension as well.

Step 4: Set Up an Emulator or Connect a Device

1. Android Emulator:
   - Install Android Studio:
     - Download and install Android Studio.
     - Open Android Studio and go to "Configure" > "AVD Manager" to create an Android Virtual Device (emulator).
     - Set up a device with your desired specifications and start the emulator.
   - Install Android SDK Command-line Tools:
     - In Android Studio, go to "Configure" > "SDK Manager."
     - Under "SDK Tools," check "Android SDK Command-line Tools" and click "Apply" to install.
   - Enable Virtualization:
     - Make sure virtualization is enabled in your BIOS/UEFI settings for better performance.
2. iOS Simulator (macOS only):
   - Open Xcode and go to "Preferences" > "Components."
   - Download and install the latest iOS Simulator.
   - Run the simulator via Xcode or use the open -a Simulator command in the terminal.
3. Connect a Physical Device:
   - Enable Developer Mode on your Android or iOS device.
   - Connect the device to your computer using a USB cable.
   - For Android, ensure "USB Debugging" is enabled.
   - For iOS, trust the computer when prompted on your device.

Step 5: Create a New Flutter Project

1. Create a New Project:
   - Open VS Code.
   - Open the Command Palette by pressing Ctrl+Shift+P (Cmd+Shift+P on macOS).
   - Type Flutter: New Project and press Enter.
   - Select "Application" and provide a name for your project.
   - Choose a location to save the project, and VS Code will create a new Flutter project.

Step 6: Run Your Flutter App

1. Open the Main Dart File:
   - In the VS Code explorer, navigate to the lib folder and open main.dart.
2. Run the App:
   - Open the Command Palette again (Ctrl+Shift+P or Cmd+Shift+P).
   - Type Flutter: Select Device and choose your connected device or emulator.
   - Press F5 to start debugging the app, or use the Run option from the toolbar.
   - The Flutter app should launch on the selected device or emulator.

Step 7: Troubleshooting and Additional Setup

1. Fixing Issues:
   - If you encounter any issues, run flutter doctor in the terminal to diagnose and fix problems.
   - Ensure that all tools are up to date and configured correctly.
2. Optional: Install Additional Extensions:
   - You may want to install other useful extensions like Flutter Intl, Pubspec Assist, and GitLens.

https://www.youtube.com/watch?v=VFDbZk2xhO4

Firebase Setup :

1. Create a Firebase Project • Go to Firebase Console. • Click on “Add Project” and follow the prompts to create a new project.
2. Add Firebase to Your Flutter App :

- Once your Firebase project is created, navigate to the project dashboard.
- Click on the Android or/and IOS icon or/and web to add a new app to Firebase

For Android:

1. Register Your app

- Enter your Android package name (‘com.example.flutter_app’)
- Provide a nickname (optional) and SHA-1 key (optional)
- Click “Register App

2. Download ‘google-services.json’

- Download the ‘google-services.json’ file and place it in ‘android/app

3. Configure Your Project

- Add (classpath ‘com.google.gms:google-services:4.3.15’) to ‘android/build.gradle’ in dependencies
- Open ‘android/app/build.gradle’ and add (apply plugin: ‘com.google.gms.google-services’

For iOS:

1. Register Your app

- Enter your iOS bundle ID in ‘project.pbxproj’ (ios file)
- Provide a nickname and App Store ID (optional)
- Click “Register App

2. Download ‘GoogleService-Info.plist’

- Download the ‘GoogleService-Info.plist’ file and place it in ‘ios/Runner’ directory

For Web:

For web just get your unique apiKey, authDomain, ProjectId, StorageBucket, messagingSenderId and appId and place them in main function like to image below to initialize Firebase

3. Add Firebase SDK to Your Flutter Project:

- In the root of your Flutter project, open ‘pubspec.yaml’
- Add the following dependencies to your project (make sure always to make the latest version)

Run ‘flutter pub get’ (or ctrl s in vscode) to install the dependencies

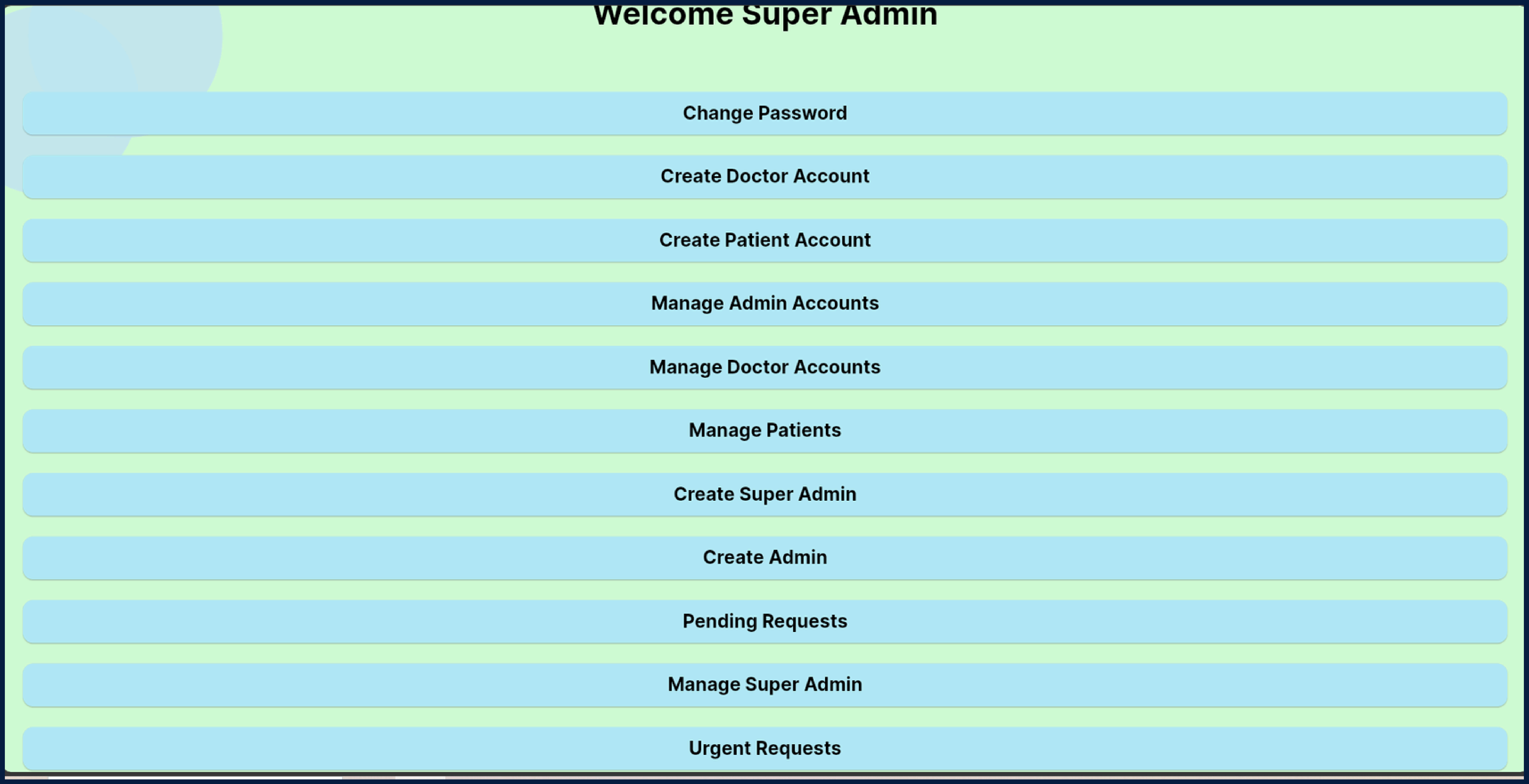
Django Setup

- Install ‘python3’
- Install ‘pip’: Open terminal and type this command ‘python -m pip install -U pip’
- Install Virtual environment : Type this command ‘pip install virtualenv • Create a virtual environment with this command ‘virtualenv env_site
- Change the current repository to env_site by ‘cd env_site’
- Access Scripts repository ‘cd scripts’
- Activate virtual environment ‘Activate’
- Install Django ‘pip install Django’
- Return to env_site ‘cd ..’
- Start a new project by this command ‘django-admin startproject mydoc’
- Access mydoc repository ‘cd mydoc’ • Run the server by typing this command ‘python manage.py runserver’ and enter the link provided by this command “http://127.0.0.1:8000/” which is the server
- Add ‘mydoc’ (or the name of your repository in ‘setting.py’ Installed Apps

°Import os and then change the ‘DIRS’ in Templates

USER MANUAL

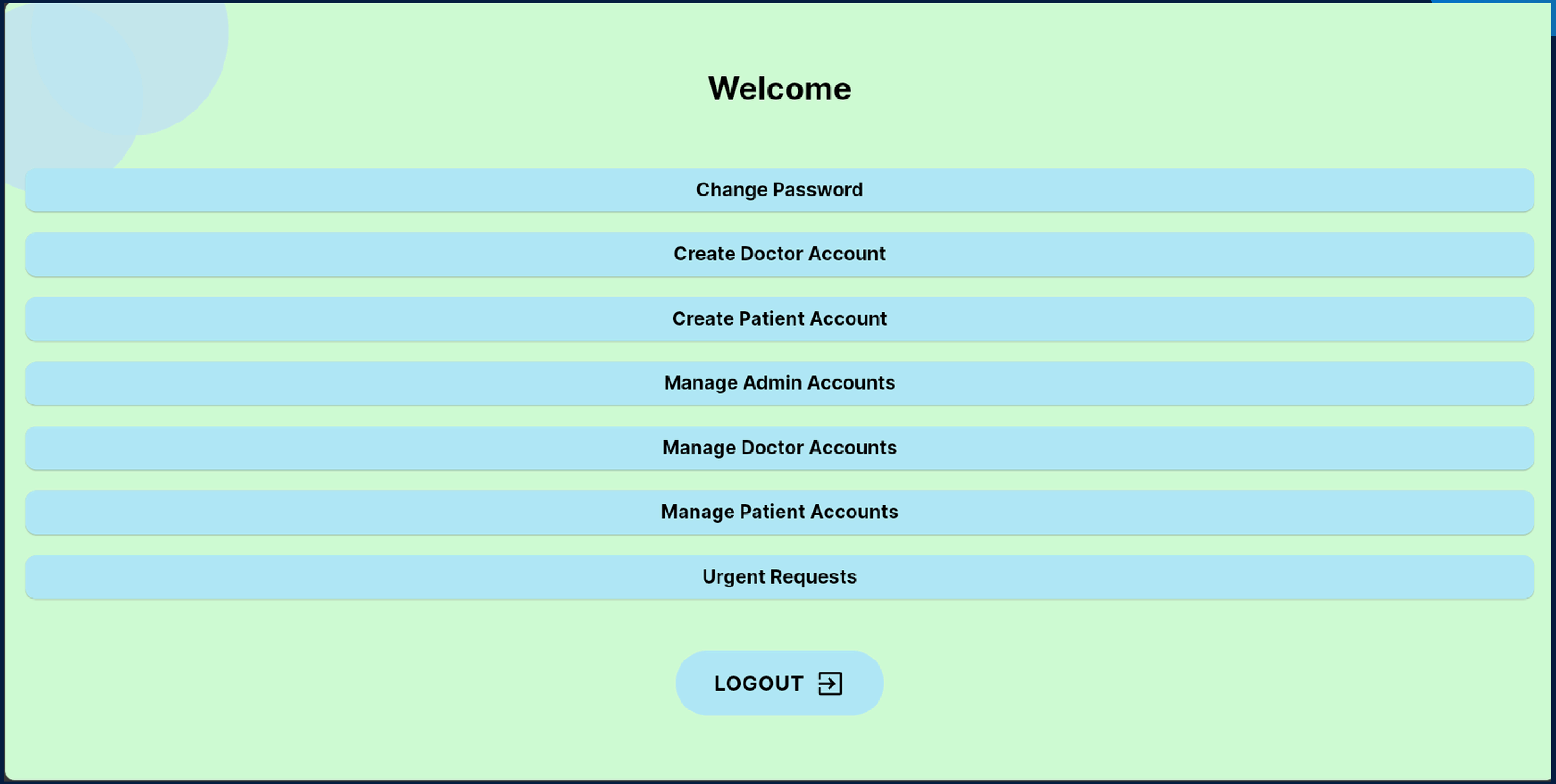
These features are designed to give the ***Super Admin*** full control over the management of users and administrative tasks within the system.



Welcome Super Admin

1. Change Password: Allows the Super Admin to change their account password.
2. Create Doctor Account: Provides the option to create new accounts for doctors in the system.
3. Create Patient Account: Enables the Super Admin to create new accounts for patients.
4. Manage Admin Accounts: Allows the Super Admin to manage existing admin accounts, including editing or deleting them.
5. Manage Doctor Accounts: Provides the ability to manage accounts for doctors, including updates, deletions, or other administrative actions.
6. Manage Patients: Allows the Super Admin to manage patient accounts, including updating personal details or removing accounts or uploading images.
7. Create Super Admin: Gives the option to create another Super Admin account with similar privileges.
8. Create Admin: Enables the creation of new Admin accounts with specific roles and permissions.
9. Pending Requests: Displays requests that are pending action, such as new account approvals or other system tasks that require admin intervention.
10. Manage Super Admin: Provides options to manage other Super Admin accounts, such as editing or revoking privileges.
11. Urgent Requests: Displays requests or tasks that need immediate attention, likely prioritized over other pending tasks.

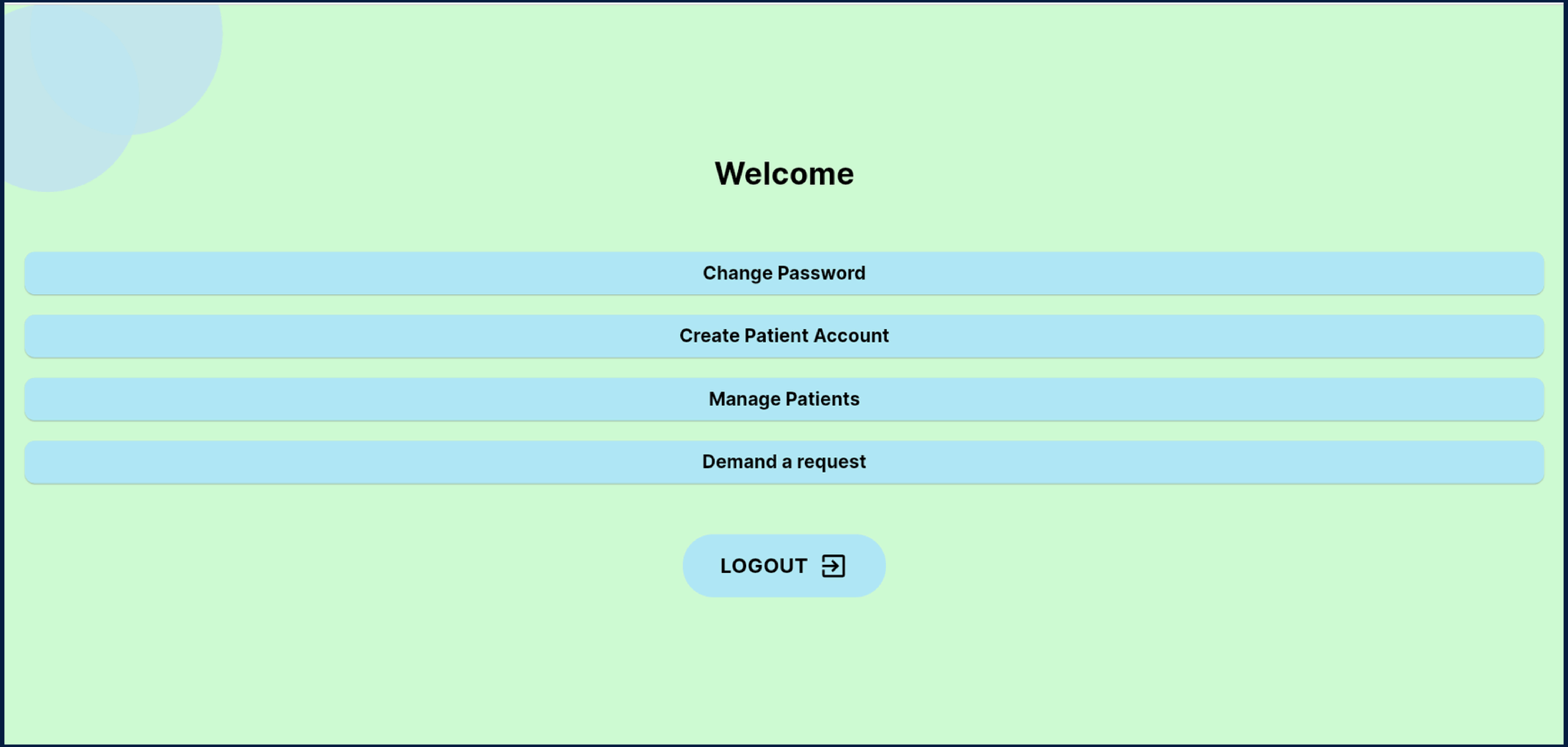
This dashboard provides essential account management functionalities for *Admin*, but it does not include some of the advanced features available to the Super Admin in the previous version.



Admin Features description:

- Change Password: Allows the user to change their account password.
- Create Doctor Account: Enables the user to create new accounts for doctors within the system.
- Create Patient Account: Provides the option to create new patient accounts.
- Manage Admin Accounts: Allows the user to manage existing admin accounts, which might include editing or deleting them.
- Manage Doctor Accounts: Enables the user to manage doctor accounts, such as updating details or removing accounts.
- Manage Patient Accounts: Allows the user to manage patient accounts, including updating information or removing them or uploading images.
- Urgent Requests: Displays any urgent tasks or requests that require immediate attention from the user.
- Logout: Provides an option for the user to log out of the system.

These features are designed to give the Super Doctor full control over the management of patients and within the system.



Welcome

Change Password

Create Patient Account

Manage Patients

Demand a request

LOGOUT

Doctor Features description:

- Change Password: Allows the Doctor to change their account password.
- Create Patient Account: Enables the doctor to create new accounts for patients.
- Manage Patients: Allows the Super Admin to manage patient accounts, including updating personal details or removing accounts and uploading images.
- Demand a request : allows the   doctor to consult the patients of another doctor after his request is accepted by an admin or superadmin

Accounts :

superadmin :
john@gmail.com
azertyuiop123
doctor :
aymen@gmail.com
azertyuiop123
admin :
admin@gmail.com
azertyuiop123

☺ *THANKS FOR READING !* ☺