DermAI: Deployment Manual

Group: Thunder Buddies

Frontend (React Native with Expo)

1. Audience Definition:

- Mobile Application Developers
- System Administrators
- Full Stack Developers

2. Platform-Specific Deployment Instructions:

- **iOS:**
  - iOS 11.0+
  - Device: iPhone 6s or newer
  - Free Disk Space: 200 MB
  - Browser: Safari
- **Android:**
  - Android OS 8.0+
  - Free Disk Space: 200 MB
  - Browser: Chrome

3. Prerequisite Installation:

- Install Node.js
- Install npm or yarn
- Install Expo CLI: `npm install -g expo-cli`

4. Configuration Instructions:

- Navigate to the project directory

Create an `.env` file:

EXPO_FIREBASE_API_KEY=AIzaSyA5hufbHHPEjFKyYIlmBMIfSYmcieSK_8c
FLASK_API_URL= http://localhost:5000/

-

5. Installation Scripts:

- `cd` **code/client**
- Run `npm install` or `yarn install`
- Run npm `start`

6. Testing and Troubleshooting:

- Utilize Expo's built-in tools and React Native Debugger for testing.
- Use `console.log` for debugging within the application.

7. Deployment:

- Run `expo build:android` or `expo build:ios` for building your respective apps.
- Follow the Expo documentation for deploying to App Store or Google Play.

Backend (Flask for ML)

1. Audience Definition:

- Backend Developers
- System Administrators
- Machine Learning Engineers

2. Platform-Specific Deployment Instructions:

- **General Requirements:**
  - RAM size: 4 GB
  - Hard Disk size: 10 GB Free Disk Space
  - CPU: Duo Core, 2.4 GHz+

3. Prerequisite Installation:

- Python 3.8+
- Flask
- Libraries: NumPy, Pandas, Matplotlib, Scikit-learn, Tensorflow or PyTorch (based on the ML model)

4. Configuration Instructions:

- Navigate to the Flask application directory

Setup the environment:

bashCopy code
```
python -m venv venv
source venv/bin/activate  # For Unix/macOS
venv\\Scripts\\activate  # For Windows
```

-

Install dependencies:

```
pip install -r requirements.txt
```

-

5. Deployment Scripts or Code Snippets:

- Activate the virtual environment

Run the Flask app:

```
flask run --host=0.0.0.0
```

-

Firebase (Database, Auth, and CRUD Operations)

1. Setup:

- Create a project in Firebase Console.
- Enable Firestore Database.
- Set up Authentication methods needed.
- Create storage for files if necessary in Firebase Storage.

2. Configuration:

- Download the Firebase admin SDK config file and integrate it into your backend service.

Set environment variables for Firebase in your Flask `.env` file
apiKey: "AIzaSyA5hufbHHPEjFKyYllmBMIfSYmcieSK_8c",

authDomain: "dermai-297f5.firebaseapp.com",

projectId: "dermai-297f5",

storageBucket: "dermai-297f5.appspot.com",

messagingSenderId: "703364639202",

appId: "1:703364639202:web:6af157239a8959aec3b1bb"

3. Security:

- Implement security rules in Firestore to manage data access.
- Use Firebase Authentication to secure endpoints in your Flask application.

Final Steps:

- Ensure all components are tested individually and in integration.
- Monitor the application closely after deployment and address any real-time issues.
- Keep dependencies updated and secure.