

# CalorieMate Deployment Manual

This document outlines the steps to set up, build, and deploy the **CalorieMate** React Native application using **Expo** and **EAS Build**.

## 1. Prerequisites

Ensure the following are installed and configured:

- Node.js (v16.x or later)
- Yarn (recommended) or npm
- Expo CLI: `npm install --global expo-cli`
- EAS CLI: `npm install --global eas-cli`
- A valid Expo account: <https://expo.dev/signup>
- An Apple Developer account (for iOS builds)
- A Google Play Developer account (for Android builds)

## 2. Project Setup

Clone the repository and install dependencies:

```
https://github.com/htmw/2025S-The-Innovators/tree/main  
cd 2025S-The-Innovators/Code/frontend  
yarn install
```

Start the local development server:

```
expo start
```

## 3. Environment Variables

Create a `.env` file in the project root to securely manage API keys:

```
HUGGING_FACE_API_KEY=your_huggingface_key  
NUTRITION_APP_ID=your_nutritionix_app_id  
NUTRITION_API_KEY=your_nutritionix_api_key
```

Ensure these are loaded using `expo-constants` or `react-native-dotenv`.

## 4. Running Locally

### iOS

```
expo start --ios
```

Requires Xcode and an iOS simulator or physical device with Expo Go.

### Android

```
expo start --android
```

Requires Android Studio emulator or a physical device with Expo Go.

## 5. Testing

Run unit and component tests:

```
yarn test
```

Lint the codebase:

```
yarn lint
```

## 6. Development Deployments (Expo Go)

To publish for internal testing via Expo Go:

```
expo publish --release-channel development
```

This pushes an OTA (Over-the-Air) update accessible in the Expo Go app.

## 7. Production Builds (EAS Build)

### Initial Setup

```
eas login  
eas build:configure
```

### Build for iOS

```
eas build --platform ios
```

### Build for Android

```
eas build --platform android
```

Build artifacts will be available at: <https://expo.dev/accounts>

## 8. OTA Updates

After deploying a production build, future code updates can be pushed without a full rebuild:

```
expo publish --release-channel production
```

Ensure your app is configured to use the correct `releaseChannel` in `app.json`.

## 9. Clean and Rebuild (if needed)

Use this if encountering build or cache-related issues:

```
rm -rf node_modules  
yarn cache clean  
yarn install  
expo start -c
```

---

## 10. Debugging

View logs from real devices:

```
expo logs
```

Restart with cache cleared:

```
expo start -c
```

## 11. App Store / Play Store Submission (Optional)

Use `eas submit` to upload builds to the respective stores:

```
eas submit --platform ios
```

```
eas submit --platform android
```

Make sure your `app.json` is properly configured with package name, display name, icons, and permissions.