Solana Wallet (Serene Wallet)

A secure, fast, and easy way to manage your Solana assets. This app allows users to create/import wallets, send/receive tokens, swap, stake, and bridge assets on the Solana blockchain.

Features

- Wallet creation and import (mnemonic/private key)
- Secure PIN and biometric authentication
- Send and receive Solana tokens
- Token swap functionality
- Staking and unstaking
- Bridge assets between chains
- Transaction history and details
- Multi-language support (English, Arabic)
- Push and local notifications
- QR code support for addresses
- Modern UI with animations and shimmer loading

Getting Started

1. Prerequisites

- **Flutter SDK**: v3.6.1 or higher
- **Dart SDK**
- **Android Studio** (for Android)
- **Xcode** (for iOS)
- **Firebase Project** (for push notifications, etc.)

---

2. Project Setup

a. Clone & Install Dependencies

```
 git clone <repo-url>
cd solana_wallet
flutter pub get
```

---

3. Firebase Configuration

a. Create Firebase Project

- Go to Firebase Console
- Add Android and iOS apps to your Firebase project

b. Add Config Files

- **Android**: Download google-services.json and place in android/app/
- **iOS**: Download `GoogleService-Info.plist` and place in `ios/Runner/`

c. Android Integration

- Ensure `com.google.gms.google-services` is in android/app/build.gradle plugins
- Confirm `apply plugin: 'com.google.gms.google-services'` is present

d. iOS Integration

- Open `ios/Runner.xcworkspace` in Xcode
- Add `GoogleService-Info.plist` to Runner target

---

4. Running the App

a. Android

```
flutter run -d android
```

b. iOS

```
flutter run -d ios
```

c. Web

```
flutter run -d chrome
```

---

5. Changing App Icons

a. Prepare Icon

- Replace `assets/icons/app_icon.png` with your icon (1024x1024 recommended)

b. (Optional) Automate with flutter_launcher_icons

Uncomment and configure in pubspec.yaml:

```
dev_dependencies:
  flutter_launcher_icons: "^0.14.4"

flutter_icons:
  android: "ic_launcher"
  ios: true
  image_path: "assets/icons/app_icon.png"
```

Run:

```
flutter pub run flutter_launcher_icons:main
```

---

6. Android Key Signing (Release Builds)

a. Generate Keystore (if needed)

```
keytool -genkey -v -keystore key.jks -keyalg RSA -keysize 2048 -validity 10000 -alias awesome
```

b. Configure key.properties

Edit/create android/key.properties:

```
storePassword=your_store_password
keyPassword=your_key_password
keyAlias=awesome
storeFile=key.jks
```

c. Enable Signing in android/app/build.gradle

Uncomment and adjust:

```
signingConfigs {
    release {
        keyAlias keystoreProperties['keyAlias']
        keyPassword keystoreProperties['keyPassword']
        storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
        storePassword keystoreProperties['storePassword']
    }
}
buildTypes {
    release {
        signingConfig signingConfigs.release
    }
}
```

---

7. iOS Signing (Release Builds)

- Open `ios/Runner.xcworkspace` in Xcode
- Go to Runner target > Signing & Capabilities
- Set your Team and provisioning profile

---

8. Customization

- **App Name**: Edit in `android/app/src/main/AndroidManifest.xml`, `ios/Runner/Info.plist`, and in `lib/core/config/app_config.dart` (change `appName`)
- **API URL**: Edit `apiEndPoint` and `domain` in `lib/core/config/app_config.dart`
- **Package Name**: Change in android/app/build.gradle and `android/app/src/main/AndroidManifest.xml`
- **Bundle Identifier (iOS)**: Change in Xcode > Runner target > General

---

9. Running Tests

```
flutter test
```

Add more tests in the `test/` directory.

---

10. Troubleshooting

- **Dependencies**: Run `flutter pub get` after any pubspec changes
- **Firebase Issues**: Double-check config file placement and Firebase Console settings
- **Icon Not Updating**: Clean build (`flutter clean`) and rebuild

---

11. Contribution

- Fork, branch, code, test, and PR.

Summary

This documentation covers setup, Firebase, running, icon change, signing, and customization for the Solana Wallet Flutter project. You can now build, run, and customize the app for your needs.