# Ainance ZIP Folder Structure (Overview)

Ainance\_Package/

├── android\_flutter\_app/

│ ├── lib/

│ │ ├── main.dart

│ │ ├── screens/

│ │ │ ├── login\_screen.dart

│ │ │ ├── dashboard\_screen.dart

│ │ │ └── settings\_screen.dart

│ │ └── widgets/

│ │ ├── custom\_button.dart

│ │ └── input\_field.dart

│ ├── pubspec.yaml

│ └── android/, ios/, assets/, build/ (auto-generated)

│

├── desktop\_electron\_app/

│ ├── main.js

│ ├── index.html

│ ├── renderer.js

│ ├── package.json

│ └── src/

│ ├── dashboard.js

│ ├── login.js

│ └── settings.js

│

├── assets/

│ ├── ainance\_logo.png

│ ├── splash\_screen.png

│ └── icons/

│ ├── icon\_round.png

│ └── icon\_square.png

│

├── LICENSE.txt

├── README.txt

└── ainance\_build\_guide.pdf # Build instructions (to be drafted here next)

# Next Step:

# I will now begin pasting the contents of main.dart and main.js next.

---

## android\_flutter\_app/lib/main.dart

```dart

import 'package:flutter/material.dart';

import 'screens/login\_screen.dart';

void main() {

runApp(const AinanceApp());

}

class AinanceApp extends StatelessWidget {

const AinanceApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'Ainance',

theme: ThemeData.dark(),

home: const LoginScreen(),

);

}

}

```

# Next: I will paste login\_screen.dart for authentication UI.

## android\_flutter\_app/lib/screens/login\_screen.dart

```dart

import 'package:flutter/material.dart';

import '../widgets/custom\_button.dart';

import '../widgets/input\_field.dart';

import 'dashboard\_screen.dart';

class LoginScreen extends StatefulWidget {

const LoginScreen({super.key});

@override

State<LoginScreen> createState() => \_LoginScreenState();

}

class \_LoginScreenState extends State<LoginScreen> {

final TextEditingController emailController = TextEditingController();

final TextEditingController passwordController = TextEditingController();

void handleLogin() {

// Replace with actual verification logic or API call

if (emailController.text == 'collins.nyandoro1@gmail.com' &&

passwordController.text == 'Nyandoro1@.') {

Navigator.pushReplacement(

context,

MaterialPageRoute(builder: (context) => const DashboardScreen()),

);

} else {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text('Invalid email or password')),

);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.stretch,

children: [

const Text(

'Welcome to Ainance',

style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

textAlign: TextAlign.center,

),

const SizedBox(height: 40),

InputField(

controller: emailController,

hintText: 'Email Address',

obscureText: false,

),

const SizedBox(height: 20),

InputField(

controller: passwordController,

hintText: 'Password',

obscureText: true,

),

const SizedBox(height: 30),

CustomButton(

text: 'Login',

onPressed: handleLogin,

),

],

),

),

);

}

}

```

# Next: I'll add the InputField and CustomButton widget implementations.

## android\_flutter\_app/lib/widgets/input\_field.dart

```dart

import 'package:flutter/material.dart';

class InputField extends StatelessWidget {

final TextEditingController controller;

final String hintText;

final bool obscureText;

const InputField({

super.key,

required this.controller,

required this.hintText,

this.obscureText = false,

});

@override

Widget build(BuildContext context) {

return TextField(

controller: controller,

obscureText: obscureText,

decoration: InputDecoration(

hintText: hintText,

border: const OutlineInputBorder(),

filled: true,

fillColor: Colors.white10,

),

);

}

}

```

## android\_flutter\_app/lib/widgets/custom\_button.dart

```dart

import 'package:flutter/material.dart';

class CustomButton extends StatelessWidget {

final String text;

final VoidCallback onPressed;

const CustomButton({

super.key,

required this.text,

required this.onPressed,

});

@override

Widget build(BuildContext context) {

return ElevatedButton(

onPressed: onPressed,

style: ElevatedButton.styleFrom(

padding: const EdgeInsets.symmetric(vertical: 16),

textStyle: const TextStyle(fontSize: 16),

),

child: Text(text),

);

}

}

```

# Next: I’ll add the dashboard and settings screen stubs.

## android\_flutter\_app/lib/screens/dashboard\_screen.dart

```dart

import 'package:flutter/material.dart';

import 'settings\_screen.dart';

class DashboardScreen extends StatelessWidget {

const DashboardScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Ainance Dashboard'),

actions: [

IconButton(

icon: const Icon(Icons.settings),

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => const SettingsScreen()),

);

},

)

],

),

body: const Center(

child: Text(

'Welcome to your trading dashboard!',

style: TextStyle(fontSize: 18),

),

),

);

}

}

```

## android\_flutter\_app/lib/screens/settings\_screen.dart

```dart

import 'package:flutter/material.dart';

class SettingsScreen extends StatelessWidget {

const SettingsScreen({super.key});

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Settings'),

),

body: const Center(

child: Text('App settings will be configured here.'),

),

);

}

}

```

ELECTRON APP CORE FILES

## desktop\_electron\_app/main.js

```javascript

const { app, BrowserWindow } = require('electron');

const path = require('path');

function createWindow() {

const win = new BrowserWindow({

width: 1000,

height: 700,

webPreferences: {

preload: path.join(\_\_dirname, 'renderer.js')

}

});

win.loadFile('index.html');

}

app.whenReady().then(() => {

createWindow();

app.on('activate', function () {

if (BrowserWindow.getAllWindows().length === 0) createWindow();

});

});

app.on('window-all-closed', function () {

if (process.platform !== 'darwin') app.quit();

});

```

## desktop\_electron\_app/index.html

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Ainance Desktop</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>Welcome to Ainance</h1>

<div id="app"></div>

<script src="renderer.js"></script>

</body>

</html>

```

## desktop\_electron\_app/renderer.js

```javascript

document.addEventListener('DOMContentLoaded', () => {

const appDiv = document.getElementById('app');

appDiv.innerHTML = '<p>Please log in to begin.</p>';

});

```

# ✅ Core Electron files added. Next: src/login.js, src/dashboard.js, and src/settings.js

## desktop\_electron\_app/src/login.js

```javascript

export function showLogin(container) {

container.innerHTML = `

<h2>Login</h2>

<input type="text" id="email" placeholder="Email" />

<input type="password" id="password" placeholder="Password" />

<button onclick="loginUser()">Login</button>

<p id="login-message"></p>

`;

}

window.loginUser = function () {

const email = document.getElementById('email').value;

const password = document.getElementById('password').value;

const message = document.getElementById('login-message');

if (email === 'admin@ainance.com' && password === 'admin123') {

message.textContent = 'Login successful!';

import('./dashboard.js').then((mod) => mod.showDashboard(document.getElementById('app')));

} else {

message.textContent = 'Invalid credentials.';

}

};

```

## desktop\_electron\_app/src/dashboard.js

```javascript

export function showDashboard(container) {

container.innerHTML = `

<h2>Dashboard</h2>

<p>Welcome to the Ainance Dashboard.</p>

<button onclick="goToSettings()">Go to Settings</button>

`;

}

window.goToSettings = function () {

import('./settings.js').then((mod) => mod.showSettings(document.getElementById('app')));

};

```

## desktop\_electron\_app/src/settings.js

```javascript

export function showSettings(container) {

container.innerHTML = `

<h2>Settings</h2>

<p>Manage your preferences here.</p>

<button onclick="goToDashboard()">Back to Dashboard</button>

`;

}

window.goToDashboard = function () {

import('./dashboard.js').then((mod) => mod.showDashboard(document.getElementById('app')));

};

```

# ✅ All Electron app source modules completed. Next: package.json and PDF build instructions.

## desktop\_electron\_app/package.json

```json

{

"name": "ainance-desktop",

"version": "1.0.0",

"description": "Ainance trading desktop application",

"main": "main.js",

"scripts": {

"start": "electron ."

},

"devDependencies": {

"electron": "^27.0.0"

}

}

```

## android\_flutter\_app/pubspec.yaml

```yaml

name: ainance\_flutter

version: 1.0.0+1

description: Ainance shared trading app for Android

environment:

sdk: ">=2.17.0 <3.0.0"

dependencies:

flutter:

sdk: flutter

cupertino\_icons: ^1.0.2

dev\_dependencies:

flutter\_test:

sdk: flutter

flutter:

uses-material-design: true

assets:

- assets/ainance\_logo.png

- assets/splash\_screen.png

- assets/icons/icon\_round.png

- assets/icons/icon\_square.png

```

## ainance\_build\_guide.pdf (Plain Text Content)

```

Ainance App Build Guide

========================

This guide helps you build the Ainance app for both Android and Windows.

---

Android APK

-----------

1. Install Flutter SDK and Android Studio

2. Open `android\_flutter\_app` in Android Studio

3. Run `flutter pub get`

4. Connect an Android device or emulator

5. Press Run (or run `flutter build apk` to generate an APK)

Windows EXE (Electron)

----------------------

1. Install Node.js and npm

2. Open terminal in `desktop\_electron\_app`

3. Run `npm install`

4. Start app with `npm start`

5. (Optional) Use `electron-builder` to create EXE installer

---

Note:

\* Use real M-Pesa logic only in production

\* Login credentials: admin@ainance.com / admin123

\* You can customize the dashboard and branding in both versions

```

# ✅ All packaging & build instructions added.