

# Capsa Platform - Technical Documentation

## 1. Introduction

### Purpose

This document serves as a comprehensive guide for developers working on the Capsa Platform, a Flutter-based financial application supporting multi-currency transactions and user role management.

## 2. Project Structure

### Directory Layout

lib/

- |— admin/      # Admin-specific functionality
- |— common/     # Shared constants and utilities
- |— functions/   # Common utility functions like api call functionalities, implementing encryption and other common functions
- |— investor/    # Investor-specific features
- |— vendor-new/   # vendor-specific features
- |— anchor/     # Anchor-specific features
- |— models/     # Data models
- |— signup/     # A very crucial directory that is responsible for logging users according to their role and also responsible for onboarding new users
- |— pages/      # Investor and vendor common pages, specially utilities for the accounts page in both the roles
- |— providers/   # State management
- |— widgets/    # Reusable UI components

SOME EXTREMELY IMPORTANT FILES :

1. **Signup/functions/login\_function.dart** : responsible for directing to pages during login, if there is any discrepancy regarding login, start debugging from this file, it

handles all the conditions that routes to specific pages, afterSuccess() being the most important function in this file

2. **Vendor-new/vendor\_new.dart, investor/investor\_new.dart, signup/signup.dart** – handles all the route data for the respective pages, whenever adding new page, add route in these files accordingly

3. **ADMIN :**

Whenever adding new page in admin side make changes to these files accordingly

- i. Admin/screens/home\_view\_big.dart : add permission under permissionsTotal variable
- ii. Admin/common/admin\_destinations.dart : add tab data using NavigationRailDestination widget under destinations total in getAdminDestinations() functions
- iii. Admin/ common/widget\_list.dart : add the new page under desktopWidgetList

NOTE : make sure that the order should always be maintained in these files, as they are mapped according to the order they are mentioned, so if you are adding a permission at 4<sup>th</sup> position, this order should be followed in the other files too

### 3. Some commonly occurring issue and how to tackle them

Issue	Probable cause	How to tackle
User not being logged in on first try	Page is getting reloaded before isAuthenticated is being set to true	Try to change the delay in the login_function.dart file
User getting routed to some page it is not supposed to just after login	There is some issue with condition handling in the afterSuccess function	First see what login data you are receiving from backend and go through the conditions
Data is being doubled on any list after each reload	The List responsible for handling the data is not being set to empty before appending data	<code>List&lt;DataType&gt; xyz = []</code> <code>xyz.addAll(response[list])</code>
Not being able to update data after an action from a dialog box	Context not being handled properly	Check how context is being passed on
Page throwing null exception even after receiving data from backend	Null conditions have not been handled properly	Handle null values while assigning them to respective models
PDF not being shown even after receiving proper link	<ol style="list-style-type: none"><li>1. CORS issue from backend</li><li>2. There is some issue with the script in index.html</li></ol>	After making sure the file is opening properly, check the index.html file
Page keeps on loading even after receiving data	There is some issue while parsing the response.body	Check the provider function responsible for handling the api call for the specific function

Double api calls occurring after a button is clicked	<ol style="list-style-type: none"> <li>1. The state of the button is not updated after first click, allowing user to trigger another click</li> <li>2. There is a known issue with https package where at times such calls can happen randomly</li> </ol>	<ol style="list-style-type: none"> <li>1. The button's state should be changed the instant it is clicked to prevent double clicking</li> <li>2. Replace https package with dio globally</li> </ol>
Data not being updated even after receiving fresh response from backend	<ol style="list-style-type: none"> <li>1. State is not being managed correctly</li> <li>2. Object responsible for maintaining data is not being updated correctly</li> </ol>	<ol style="list-style-type: none"> <li>1. Confirm you are calling setState() / notifyListeners() properly</li> <li>2. Check whether the object is being updated correctly</li> </ol>

# Some important notes

## 1. Encryption:

- a. All the API payloads are encrypted using AES encryption
- b. Encryption key and IV are different for all the versions (prod, qa and dev)
- c. In **PROD and QA**, you will receive encrypted payloads for all the apis with /signup endpoints, but in **DEV** those are not encrypted
- d. You will get the encryption keys in their specific main files

## 2. Responsive UI:

- a. For responsive UI there is a custom class built that decided whether the view is in mobile or desktop mode accordingly
- b. The login can be found in **/lib/common/responsive.dart**