

## Brief Flutter Development Assignment

We've put together this short assignment to evaluate how you approach real-world Flutter app development. The focus is on your understanding of clean architecture, API integration, state management, and overall UI/UX quality.

You'll be using the [FakeStoreAPI](#) — a public REST API that provides products, carts, user accounts, and authentication — to build a mini e-commerce app.

Note:

### You're expected to use:

- MVVM architecture
  - BLoC for state management (Cubit or full BLoC)
  - Dio or http for networking
  - Clean UI/UX principles
- 

### Scenario

Imagine you've been hired to build the customer-facing frontend for an e-commerce store. The backend is already functional via the FakeStoreAPI. Your job is to build an app that allows users to browse products, login, manage their cart, and view their profile.

This is your chance to show how you structure your Flutter projects, handle real API data, and create an intuitive shopping experience.

---

### Deliverables

Your Flutter app should include:

1. **Login Functionality**
  - Authenticate using `/auth/login`
  - Handle and store tokens securely
  - Redirect to the main app screen on success
2. **Product Listing**
  - Use `/products` to fetch and display product cards with image, title, price
  - Allow filtering by category with `/products/category/:category`
3. **Product Details**
  - Tap a product to view details from `/products/:id`
  - Show full description, image, and "Add to Cart" button

#### 4. Cart Management

- View cart items via `/carts/user/:userId`
- Show item quantity, remove option, and cart total

#### 5. User Profile

- Fetch user info from `/users/:id`
- Show basic details like name, email, and address

---

#### Code Requirements

- Follow MVVM folder structure (models, viewmodels, blocs, views, services)
- Use Dio for HTTP requests
- State should be managed using BLoC or Cubit
- Handle loading, error, and empty states cleanly
- Prioritize UI responsiveness and good user experience

---

#### Bonus (Optional)

If you've got extra time:

- Add pull-to-refresh on product list
- Implement dark mode toggle
- Add a search bar for products

---

#### Submission Instructions

- Upload your code to GitHub (or zip it up)
- Include a **README** explaining:
  - How to run the app
  - What features are working
  - Screenshots or screen recordings (if possible)
- Email the link or file to the HR contact within **72 hours**