

# Trainee Program

## Phase 1: UI Recreation

- **Goal:** Learn HTML/CSS and Oracle JET basics.
  - **Activity:**
    - Recreate screens provided
    - Ensure responsive design for web screens only, follow branding/colors.
    - No functionality yet — just static pages.
  - **Deliverable:** Working HTML/CSS screens styled like the shared mockups.
- 

## Phase 2: Dynamic Rendering with Knockout

- **Goal:** Learn Knockout.js for data binding and conditional rendering.
  - **Activity:**
    - Make the CNIC/Account input fields bind to Knockout observables.
    - Add form validation (e.g., CNIC must be 13 digits).
    - Simulate flow: if CNIC entered → show "proceed" state (without API).
  - **Deliverable:** Functional frontend with Knockout-based dynamic updates.
- 

## Phase 3: Backend API Development (Spring Boot / Jersey in Java)

- **Goal:** Build REST APIs with CRUD operations.
  - **Entities:** `User`, `Account`.
  - **Activity:**
    - Create following APIs:
      1. **GET** `/api/accounts/{cnic}` → Fetch account info.
      2. **POST** `/api/accounts` → Register new account (add).
      3. **PUT** `/api/accounts/{id}` → Update login details (username/password).
  - **Deliverable:** REST API running locally, tested with Postman.
-

## Phase 4: Integration (Frontend + Backend)

- **Goal:** Connect Oracle JET + Knockout frontend with Spring Boot backend.
  - **Activity:**
    - Use AJAX/Fetch to call APIs.
    - CNIC entered in frontend → call **GET API** → fetch account details and render dynamically.
    - "Create Login Details" form → call **POST API** to register login credentials.
    - "Update Login Details" → call **PUT API** to update password.
  - **Deliverable:** Functional end-to-end integration.
- 

## Phase 5: SQL Stored Procedure

- **Goal:** Learn procedural SQL for insert/update logic.
  - **Activity:**
    - Create stored procedure `sp_create_account` that:
      - Takes CNIC, Account Number, Username, Password.
      - Inserts into `Accounts` and `Users` tables.
  - **Deliverable:** Stored procedure tested via SQL client.
- 

## Phase 6: SQL View

- **Goal:** Expose simplified read-only structure.
  - **Activity:**
    - Create a SQL View `vw_account_summary` joining:
      - User Table (CNIC, Username, Name)
      - Account Table (Account Number, Type)
  - **Deliverable:** View returning consolidated data.
- 

## Phase 7: Expose SQL View via API

- **Goal:** Return SQL View data via Spring Boot.

- **Activity:**
  - Add API endpoint:
    - **GET** `/api/accounts/summary/{cnic}` → Returns view data for user.
- **Deliverable:** Trainees can query summary data (CNIC → Account info + login details) via API.