

# Account Creation Form

## Intern Interview Task

### Objective

Build a full-stack application where users can create new Accounts by selecting:

- Entity (from existing `entities` table)
- Account Type (`AST`, `LIA`, `EQT`, `REV`, `EXP`)
- Subcategory (filtered by selected Account Type)
- Account Name
- Opening Balance

### Technology Stack

- **Frontend:** React.js
- **Backend:** Express.js (Node.js)
- **Database:** MySQL

### Database Schema

```
-- Entities Table
CREATE TABLE entities (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(250),
  NTN INT
);

-- Account Subcategories
CREATE TABLE account_subcategories (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  Account_Type ENUM('AST', 'LIA', 'EQT', 'REV', 'EXP'),
  Subcategory VARCHAR(250)
);

-- Accounts
CREATE TABLE accounts (
  ID INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(250) UNIQUE,
  Type ENUM('AST', 'LIA', 'EQT', 'REV', 'EXP'),
  Entity INT,
  Opening_Balance FLOAT DEFAULT 0,
  Opening_Date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  Balance FLOAT DEFAULT 0,
  Subcategory INT,
  FOREIGN KEY (Entity) REFERENCES entities(ID),
  FOREIGN KEY (Subcategory) REFERENCES account_subcategories(ID)
);
```

## Backend Requirements

Set up the following endpoints at your backend:

### API Endpoints

Endpoint	Method	Description
/api/ents	GET	Fetch all entities
/api/subCats?accType=<type>	GET	Fetch subcategories by account type
/api/accounts	POST	Create a new account

## Frontend Requirements

Build a React form with these fields:

- **Account Name** (text input)
- **Entity** (dropdown - /api/ents)
- **Account Type** (dropdown - static list)
- **Subcategory** (dropdown - /api/subCats?accType=<type>)
- **Opening Balance** (number input)

### Behavior

- Selecting an Account Type should dynamically fetch subcategories.
- Changing Account Type should reset Subcategory.
- All fields should be required.
- Form should be visually appealing (Recommended).
- Validate all fields before submission (Optional).
- Submit the form via POST /api/accounts.
- The /api/accounts endpoint should expect a JSON payload { name, ent, type, subCat, openBalance }.
- Use Axios or Fetch for API calls.
- Show a success or error message based on the server's response.
- Include loading indicators where necessary (Optional).

## API Convention

All endpoints must be under /api/, e.g., /api/ents, /api/accounts.

## Design Requirements

Keep the UI simple and intuitive with proper form validation and dynamic dropdowns.

## Recommendations

- The frontend should be your top priority.
- Try to get the main logic of your form working first, including setting up calls to your API endpoints.
- Whilst styling your form is recommended, it should be the last thing you do.
- If you need help with anything, feel free to ask!
- Relax and have fun!