**Group 1 – Assessment**

**Group members:**

**Experience**:

balusu.nagendra@capgemini.com

tejasvi.a.rohidas-lotlikar@capgemini.com

siddharth.c.verma@capgemini.com

**Freshers**:

kunal.b.kumar@capgemini.com

shubham-vitthal.ute@capgemini.com

dheeraj-sanjay.potdar@capgemini.com

sameer-laxminarayan.mogadpally@capgemini.com

[siddharth-laxman.more@capgemini.com](mailto:siddharth-laxman.more@capgemini.com)

**Banking Application**

**Pre-requisite:**

Install Node, Install Angular Cli, Install VS Code,

Install Json Server- npm install json-server

Start Json server - json-server --watch ApplicationDB.json

**Description:**

Create a basic banking application where users can:

* Create an account
* View account details
* Perform transactions (deposit, withdraw, transfer)
* View transaction history

**Features:**

1. **User Authentication**: Sign up and log in functionality.
2. **Account Management**: Create and manage bank accounts.
3. **Transactions**: Deposit, withdraw, and transfer funds between accounts.
4. **Transaction History**: View a history of all transactions.

**Learning Points:**

Use Auth Interceptor -- Handle unauthorized or forbidden errors (e.g., redirect to login)

Use of Observable

Pass Data from Child to parent and Parent to child

Filters, Map

Routing + (Lazy Loading)

State Management

Reactive Forms – (NgRx)

Custom Pipe

Custom Directive

RxJs Operators

Change Detection Strategy

**Note**: All the learning points should be cover with proper explanation.

**.Net: (Backend API)**

.Net Core Web API (.Net Core 6)

Entity Framework Core

Implementation of Authentication and Authorization (JWT Token)

Web API Routing

Implementation Exception Filter

Implementation of Repository Design Pattern

**.Net Assignment:**

Online Compiler - [| C# Online Compiler | .NET Fiddle](https://dotnetfiddle.net/)

Implementation of

Factory Design Pattern – Creational Design pattern

Observer Design Pattern – Behavioural Design Pattern

All Supportive functionalities should be cover in related to Banking Application with the help of model class.

**Note**: No DB operations would be needed.