

# **PROBLEM STATEMENT**

## **APPLICATION NAME - MY BANKING**

### **SCENARIOS**

#### **1. LOGIN SCREEN**

- ❖ When you launch URL, login page should be displayed which will allow the user to enter their credentials to access the account details
- ❖ If the credentials is invalid, the error message should be displayed on the screen
- ❖ Login page should also contain “Forgot Password” which will allow the user to reset the password on screen and “Register” which will allow the user to register against the account number

#### **2. DASHBOARD**

- ❖ Once the user login is successful, it allows the user to land on the dashboard which contains the account details (Account Name, Account Balance and Account Number)
- ❖ The Button “Add Payee” should be present in order to add any payee
- ❖ The Button “Remove Payee” should be present in order to remove any payee
- ❖ The Button “Money Transfer” should be present in order to transfer the money
- ❖ The link “logout” should be present at top right of the page in order to logout from the application

#### **3. ADD PAYEE**

- ❖ When the button “Add Payee” is clicked, the form to enter the details like “Payee Account Number”, “Payee Account Name” and Payee Account Bank” should be displayed. The input should be accepted once the button “Add” is clicked.

#### **4. REMOVE PAYEE**

- ❖ When the button “Remove Payee” is clicked, the form to enter the details like “Payee Account Number” and “Payee Account Bank” should be displayed. The input should be deleted once the button “Remove” is clicked.

#### 5. **MONEY TRANSFER**

- ❖ When the button “Transfer Money” is clicked, the form to enter the details like “Payee Account Number”, “Payee Account Bank”, “Amount to transfer” should be displayed. When the button “Transfer” is clicked, the required balance should be modified according to the money transferred.

#### 6. **DEPOSIT MONEY**

- ❖ When the button “Deposit Money” is clicked, the form to enter the details like “Amount to Deposit” should be displayed. When the button “Deposit” is clicked, the required balance should be modified according to the money deposited.

### **IMPLEMENTATION - HINTS**

#### **NOTE:**

This application is not end-to-end banking application, which will not include the security checks during login to “My Account”, OTPs in order to deposit/transfer money, will not retrieve IFSC code in order to transfer the money. This requirement is designed in such a way that the participant should be clear with the required libraries learned during the training program.

#### 1. **CGI Programming**

- ❖ The front-end should be designed and the implementation of each and every scenario should be done with CGI Module.
- ❖ The look and feel of the page can be enhanced by adding “CSS (Cascading Style Sheets)” but it’s not mandatory.

#### 2. **Classes and Objects**

- ❖ The required variables and methods should be declared inside the class called “Account”.
- ❖ All the activities should be of object oriented.

### 3. **Regular Expressions**

- ❖ All the validations that need to be carried out in all the forms should be done using “Regular Expressions”.

### 4. **Database Programming**

- ❖ All the details should be stored in the database
- ❖ Required tables should be created prior to launch the application and the data should be stored in appropriate tables
- ❖ Account Number should be populated in the appropriate table prior to “User Registration”, as user can be registered based only on account number

### 5. **Logging**

- ❖ All the activities should be logged in the log file “my\_account\_app.log” with the appropriate logging levels

## **BEST PRACTICES**

- ❖ Variable names should be appropriate with the value that you are going to store. If the variable name is long, the words should be separated by ‘\_’. (Eg):To store the account number, instead of having the variable name as “accountnumber”, it can be created as “account\_no” or “acc\_no”

- ❖ Function names should be appropriate with the set of actions that is going to be done, i.e., by seeing the name of the function itself, the user has to decide the set of actions that can be implemented in it. (eg): `add_payee()`, `remove_payee()`, `get_acc_balance()`, etc.,
- ❖ To validate user inputs, optimized regular expression patterns should be used.
- ❖ Database needs to be created with set of tables in which the data can be stored in the relational approach.
- ❖ Implementation of “CSS” with front end is optional and it depends on the participant whether it needs to be used or not.