# Project Title

A Simple Banking System

# Project Outline

Develop a simple Banking System where users can create bank accounts, and deposit, withdraw, and transfer money between accounts.

# Functional Requirements

1. Signup Page:
   1. Should allow unauthenticated users to create bank accounts.
   2. Users should be allowed to start with a bank balance of their choice.
   3. Should take in a password, which the user can use later to login
   4. The SignUp Page and LoginPage should be the only two pages that can be accessed by unauthenticated users
2. Home Page:
   1. The Home Page should display AccountId, current balance and past 5 transactions
3. Transfer Money Page:
   1. This page should have a form which has two fields
      1. Recipient Account Number
      2. Transfer amount
   2. A successful transaction should result in the current user’s balance being reduced by the transfer amount and the recipients.
   3. The transaction should fail if the transfer amount is more than the user’s account balance or if the recipient account number is invalid.
4. Transactions Page:
   1. This page should list all the transactions for the current user
   2. The transaction list should display:
      1. Type of Transaction (Transfer/Withdrawal/Deposit)
      2. Recipient id (only for transfers)
      3. Date and time of transaction
      4. Transaction amount
      5. Closing Balance(the net balance just after the transaction)
5. Withdraw Money Page:
   1. This page should contain a form with a single field
      1. Withdrawal Amount
   2. A successful transaction should result in the current user’s balance being reduced by the withdrawal amount.
   3. The transaction should fail if the withdrawal amount is greater than the current account balance.
6. Deposit Money Page:
   1. This page should contain a form with a single field
      1. Deposit Amount
   2. A successful transaction should result in the current user’s balance being increase by the deposit amount.

# Creative Freedom

The functional requirements define abstract functionalities. There are multiple ways to implement the functionalities discussed above. This project gives you the creative freedom to choose your data structures for the project.

Anything not mentioned in the functional requirements, but necessary to develop the complete application can be added in by the student.

The student is supposed to make educated decisions on how to place different elements for different functionalities so as to display a better UI design.

# Marking Scheme

You will be marked on the following parameters:

1. The complete application should be developed using the Microservices Architectural Style. You will be scored on the domains that you select for your Microservices.
2. Package names should confirm to the Layered Architecture or Domain Driven Design. You will be scored on your choice of package names.
3. You will be given marks for following Industry Best Practices.
4. Your APIs should be RESTful and should implement proper mappings for CRUD Operations. You will be scored for proper mappings.
5. You should use Based Authentication and Authorization in your application. You will be scored for proper implementation.
6. You will be scored for proper implementation of Zuul Proxy and Eureka Discovery Service.
7. Finally, you will be scored for all the functionalities defined in this document.

# Submission

1. Create a new STS Workspace
2. Develop all the projects in this workspace
3. Compress the workspace folder as a .zip file
4. Rename the folder to **p3\_yourFullName.zip** (If you don’t follow this naming convention, then it will become hard to score your project)
5. Submit it to the appropriate person