README

# Tech-Stack Used: ReactJS - Hooks, NodeJS, Postman, Microsoft SQL Server

INSTALL BELOW Softwares for **Windows 10**

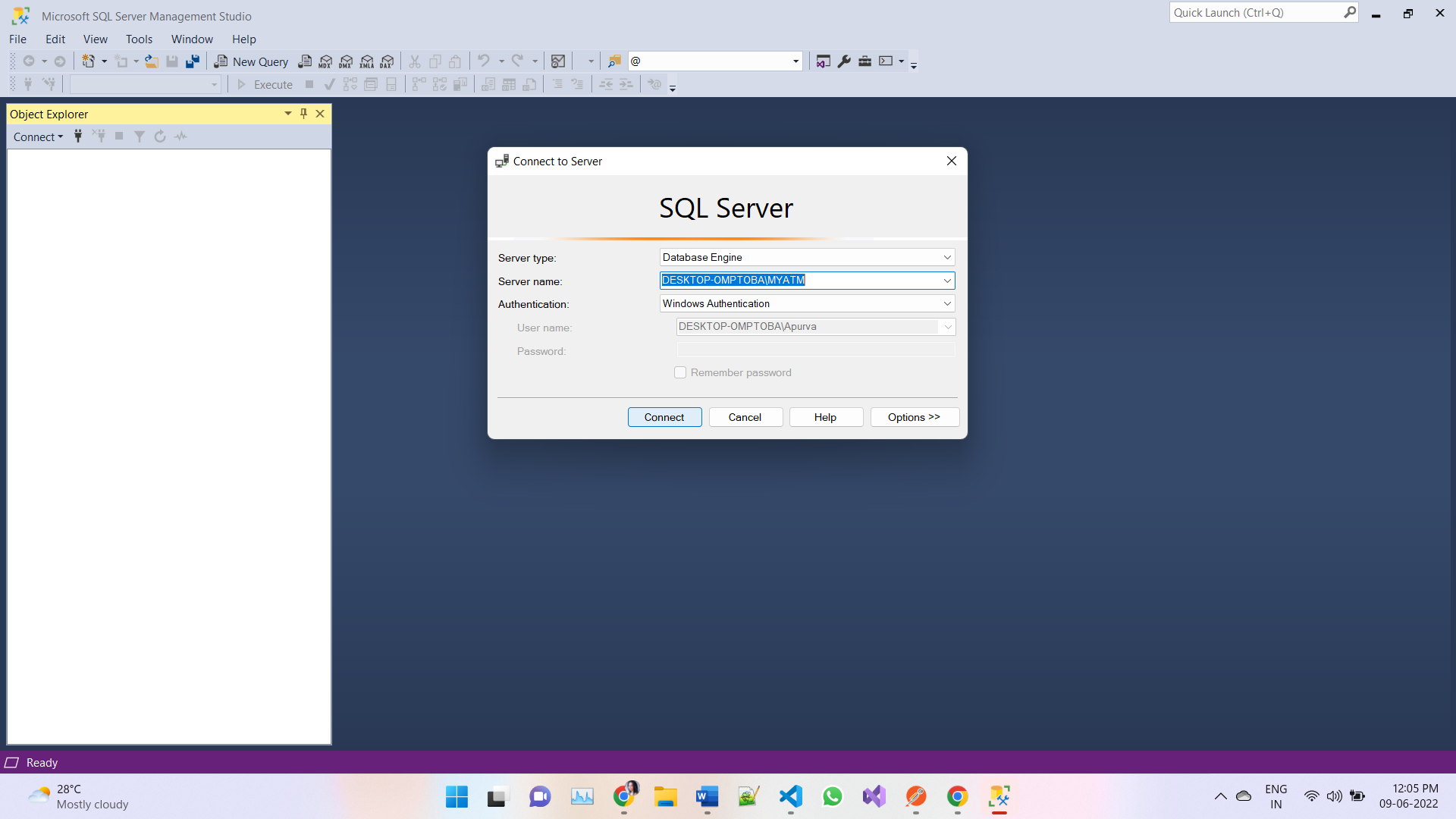
1. Install VS Code - <https://code.visualstudio.com/download>
2. Install Postman - <https://www.postman.com/downloads/>
3. Install NodeJS - <https://nodejs.org/en/download/>
4. Install SQL Server 2019 Developer – SSMS and SQL Configuration Manager - <https://www.microsoft.com/en-in/sql-server/sql-server-downloads>

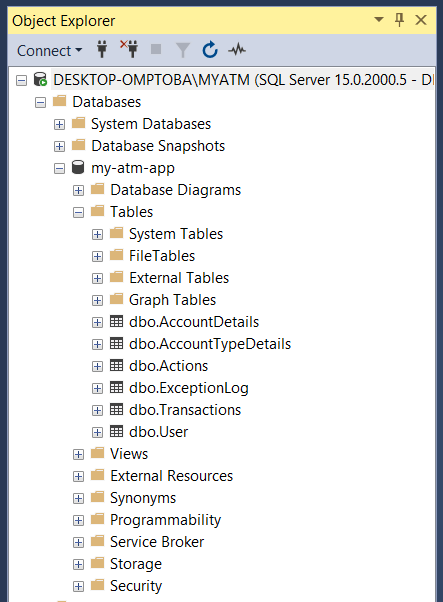
Unzip the Files named or Clone from the GIT Repository (Please try accessing the code from GITHUB if unable to - I have added the zip file too)

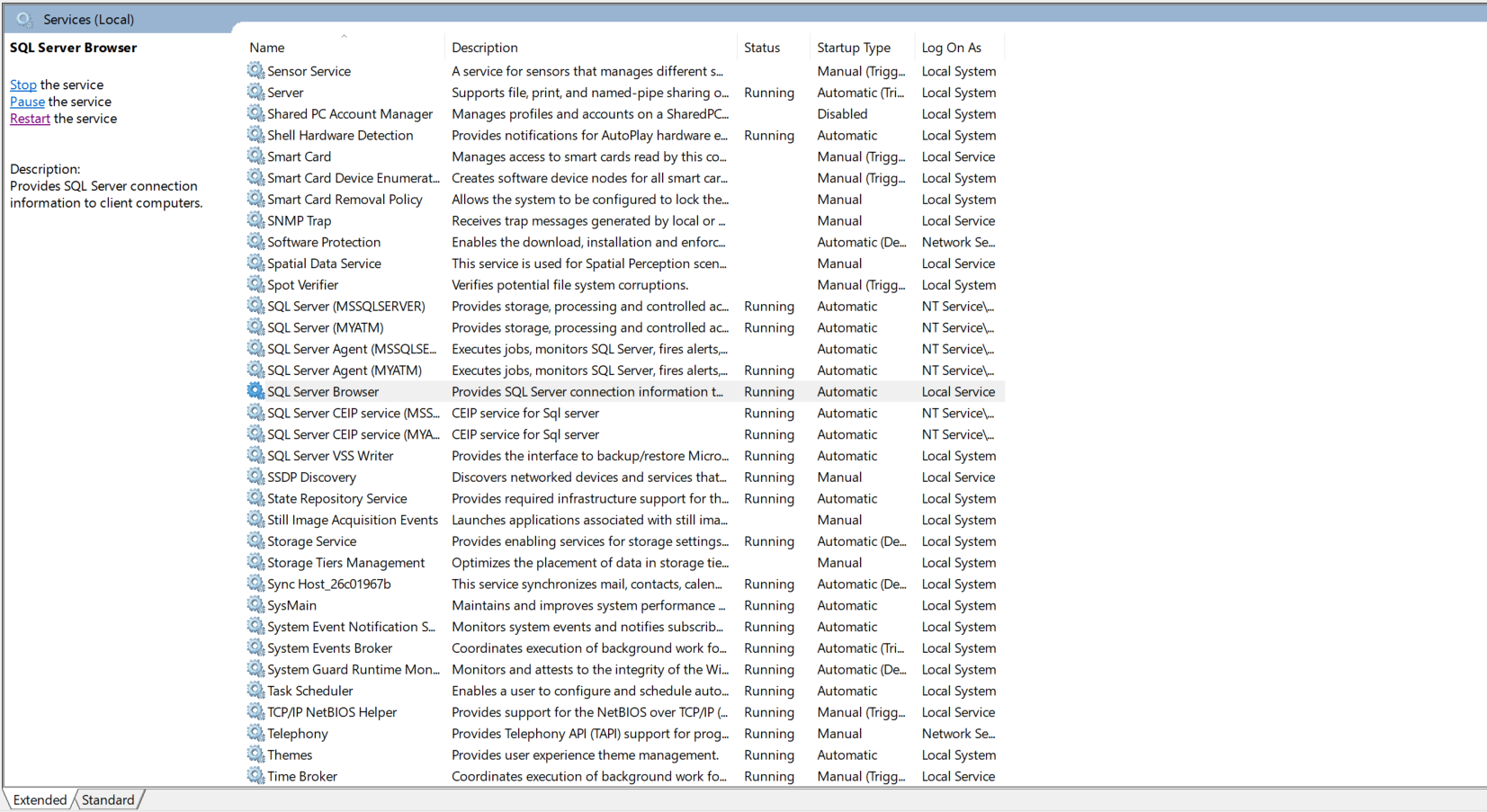
* ui-banking-app - <https://github.com/apurvatare05/ui-banking-app/tree/master>
* ui-banking-service - <https://github.com/apurvatare05/ui-banking-service/tree/master>
* FinalScript – ui-banking-sql - SQL Server Scripts - <https://github.com/apurvatare05/ui-banking-sql>

# BACKEND - SQL Server SET-UP

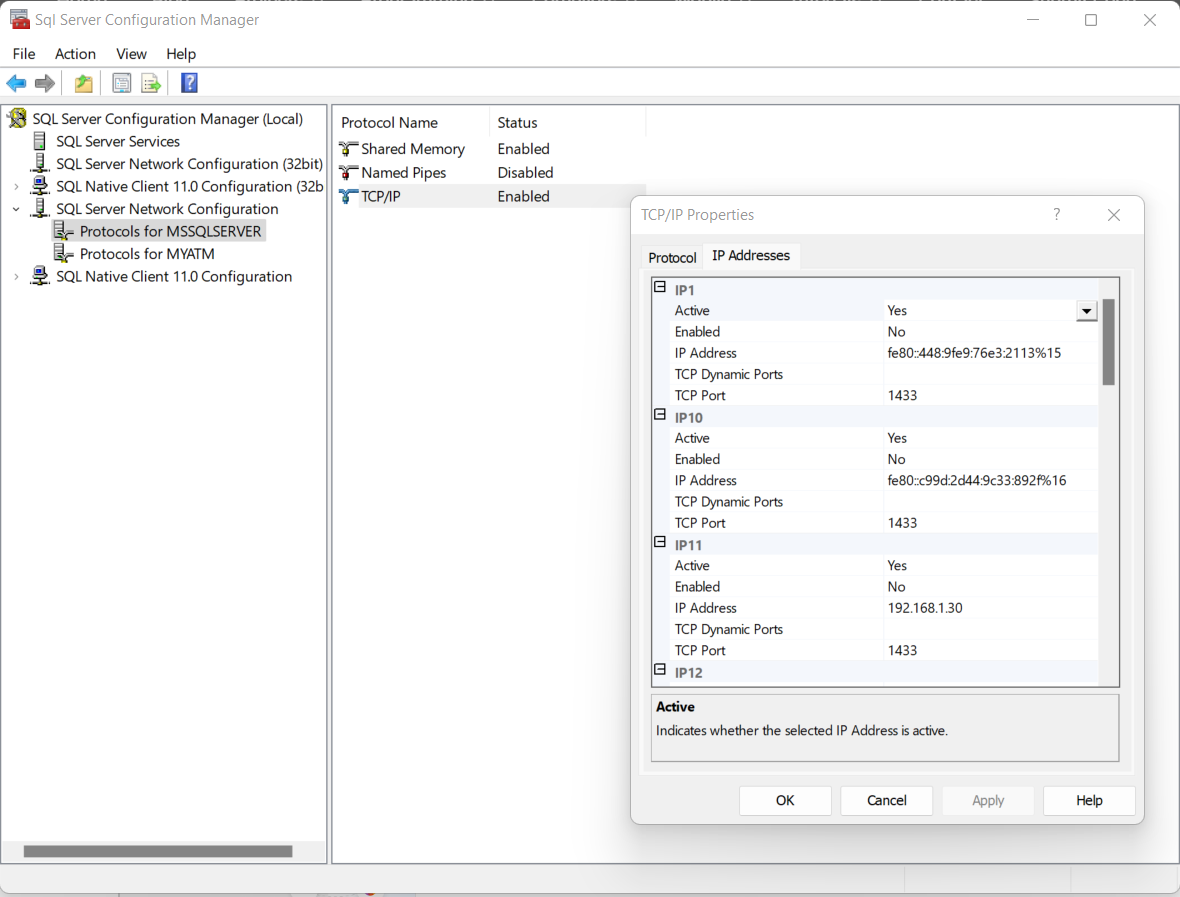
1. Make sure your SQL Server instance name is “MYATM”
2. Database Name: my-atm-app







1. In Order to enable connection of Node JS to SQL Server we need to enable – SQL Server Browser



1. Also, in order to connect NodeJS to Database server do the following steps:
   1. Open the SQL Server Configuration Manager.
   2. In SQL Server Network Configuration select Protocols for [yourServerInstance]. In the right-hand pane, make sure that TCP/IP is Enabled.
   3. Open the 1433 port in your firewall.
   4. In the SQL Server Configuration Manager, right-click TCP/IP and select Properties.
   5. Select the IP Addresses tab and make sure the TCP Port for IP1 is 1433. - If you are using an named instance, create an extra rule in your firewall with the port 1434
2. Setting up the Database
   1. Create an Instance named – MYATM
   2. Create a Database named – my-atm-app
   3. Run all the scripts provided in the attachment as per below steps
      1. Execute all the scripts named as (For Eg.) dbo.AccountDetails.Table – ending from Table
      2. Next Execute all the scripts with names (For Ex.) dbo.AddAccount.StoredProcedure – ending from StoredProcedures
   4. Now run the below queries to check if we have all the data (Also in – SELECT\_ALL\_TABLES.sql)

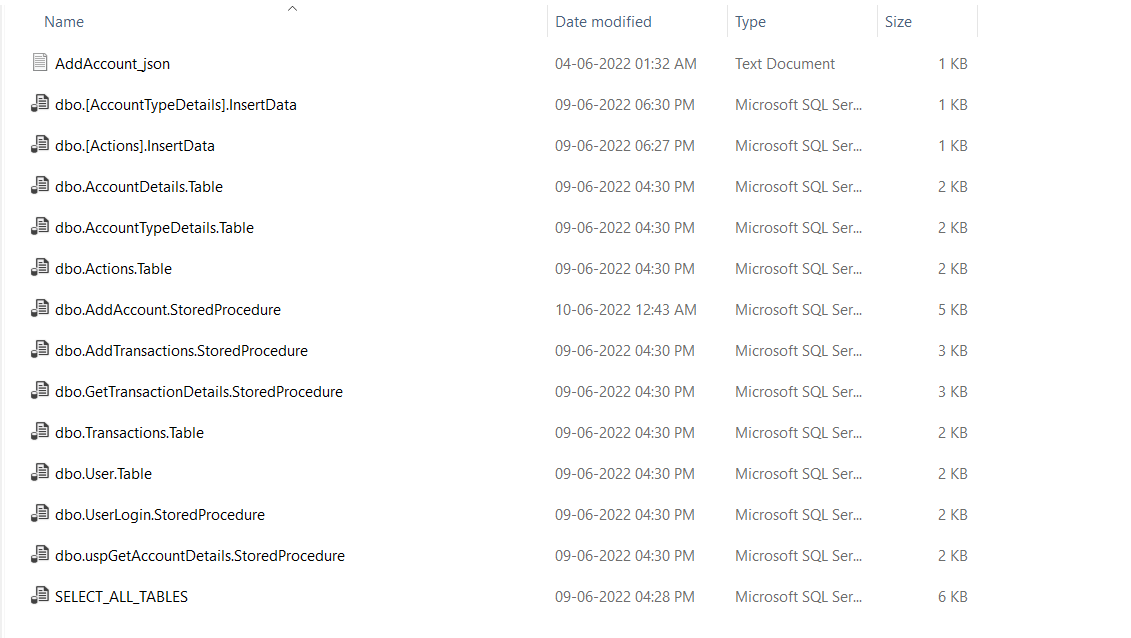
select \* from [dbo].[Transactions];

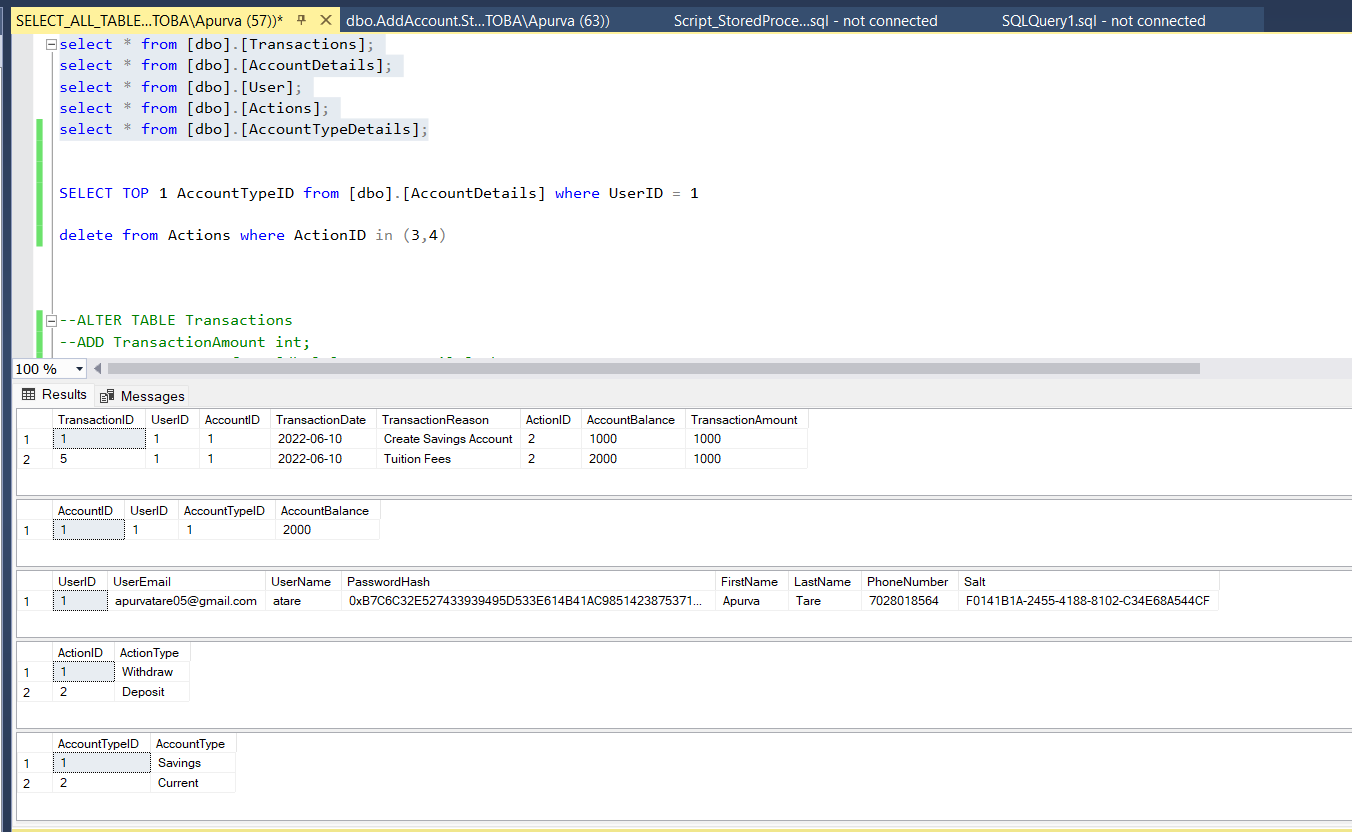
select \* from [dbo].[AccountDetails];

select \* from [dbo].[User];

select \* from [dbo].[Actions];

select \* from [dbo].[AccountTypeDetails];





If you find there is data in the User Table – I have added it so because we need it to login.

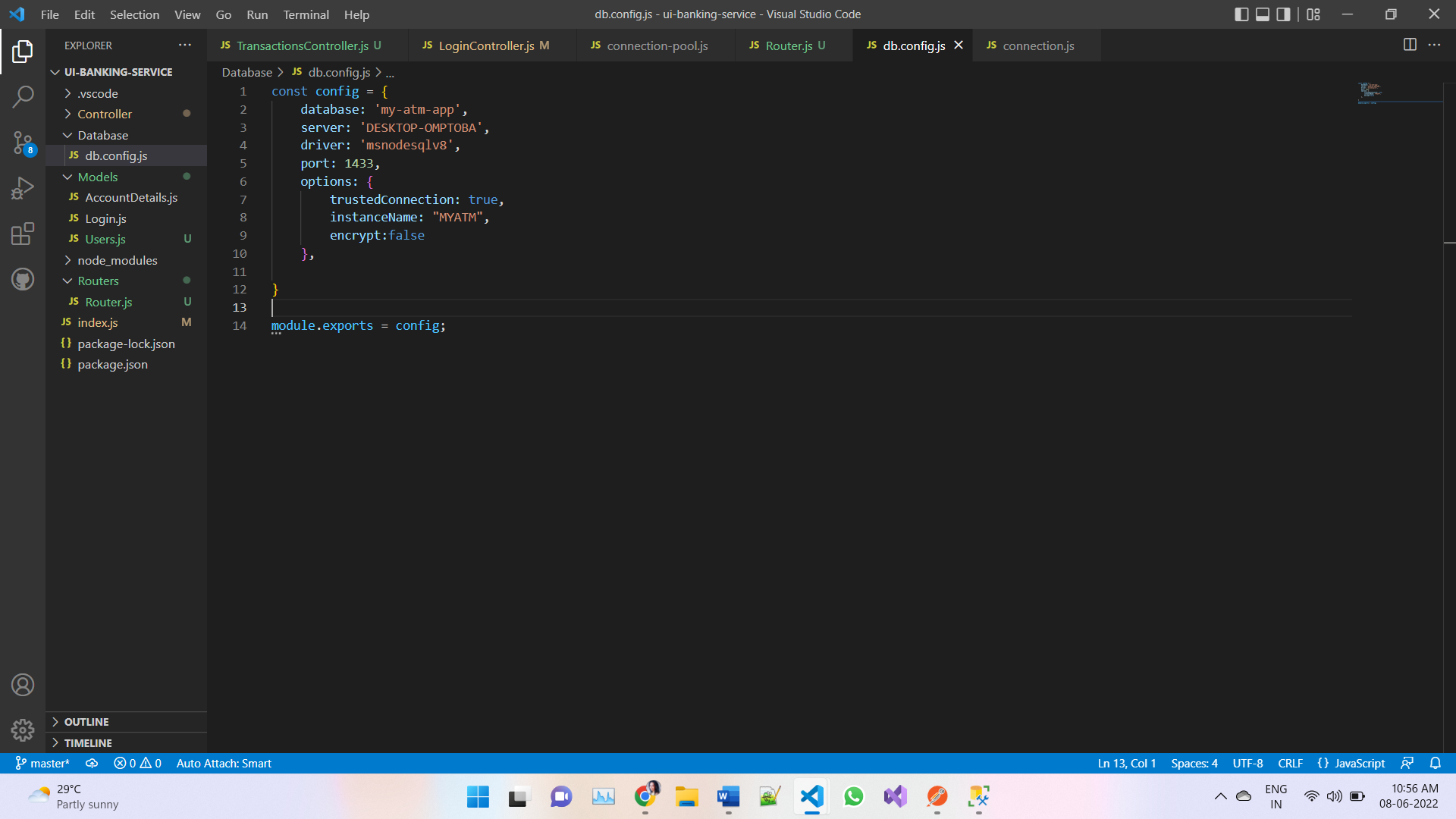
I was going to work on the Register but I didn’t get time to work on it. So during Login – Please take the email ID and password to login.

So, now the SQL Set-up is completed

# Middleware - NODE JS SET-UP

Once you unzip “ui-banking-service” you need to change the below configurations to connect to your local instance of SQL Server

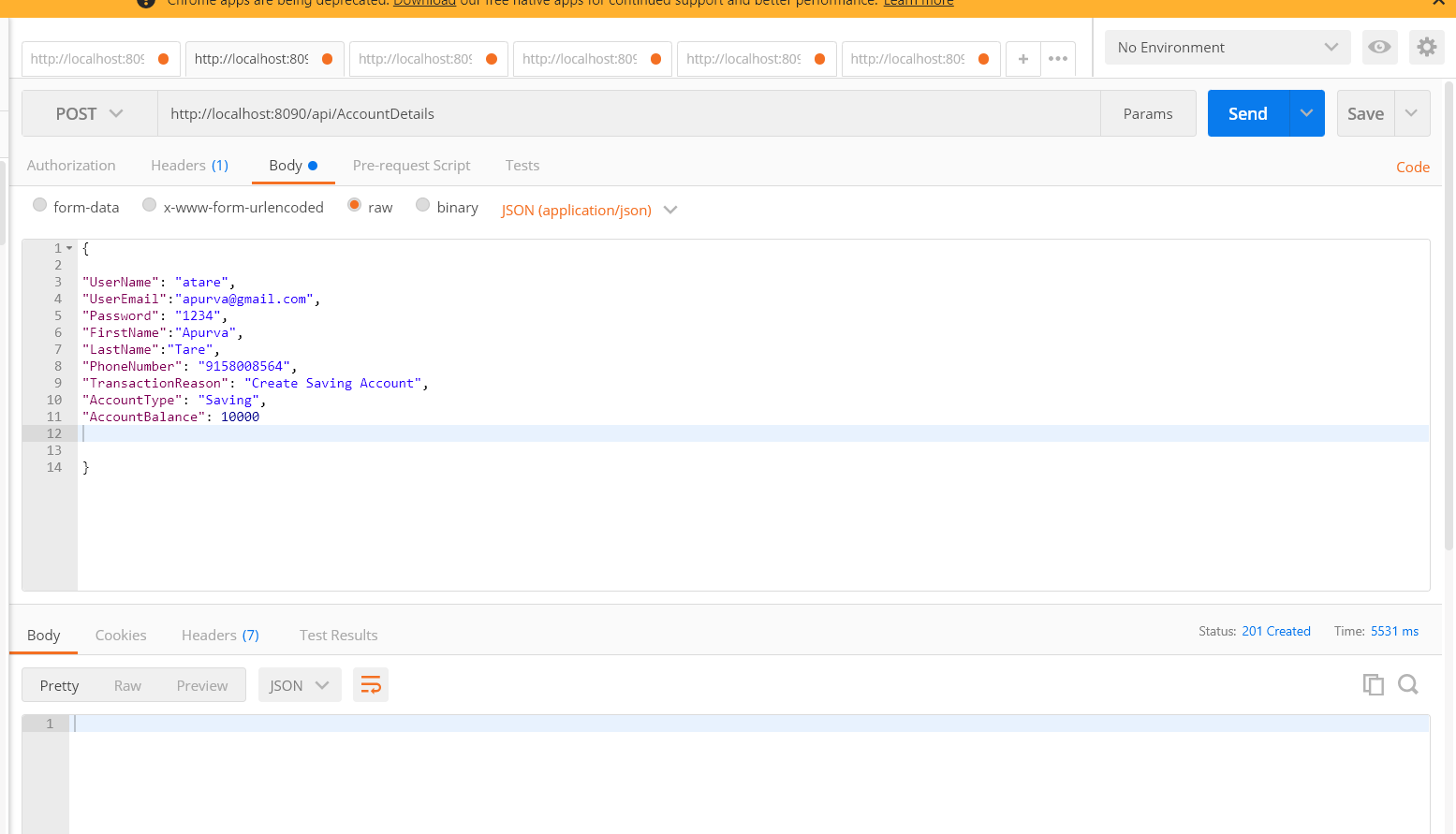
Change the **Server Name**



Once the code is downloaded for **ui-banking-service**

* Run “**npm install**” to install the node modules according to package.json
* Next you need to start the Node.js Application by running **“npm start”** in the terminal
* Once you have started the application – Start Postman and try to access the api
* The link should be like below:
  + POST: <http://localhost:8090/api/AccountDetails>
  + GET: <http://localhost:8090/api/Login?user_email=apurvatare05@gmail.com&&user_password=123>
  + POST: <http://localhost:8090/api/Transactions>
  + GET: <http://localhost:8090/api/Transactions?UserID=1&AccountID=1>
  + GET: <http://localhost:8090/api/AccountTypeDetails>

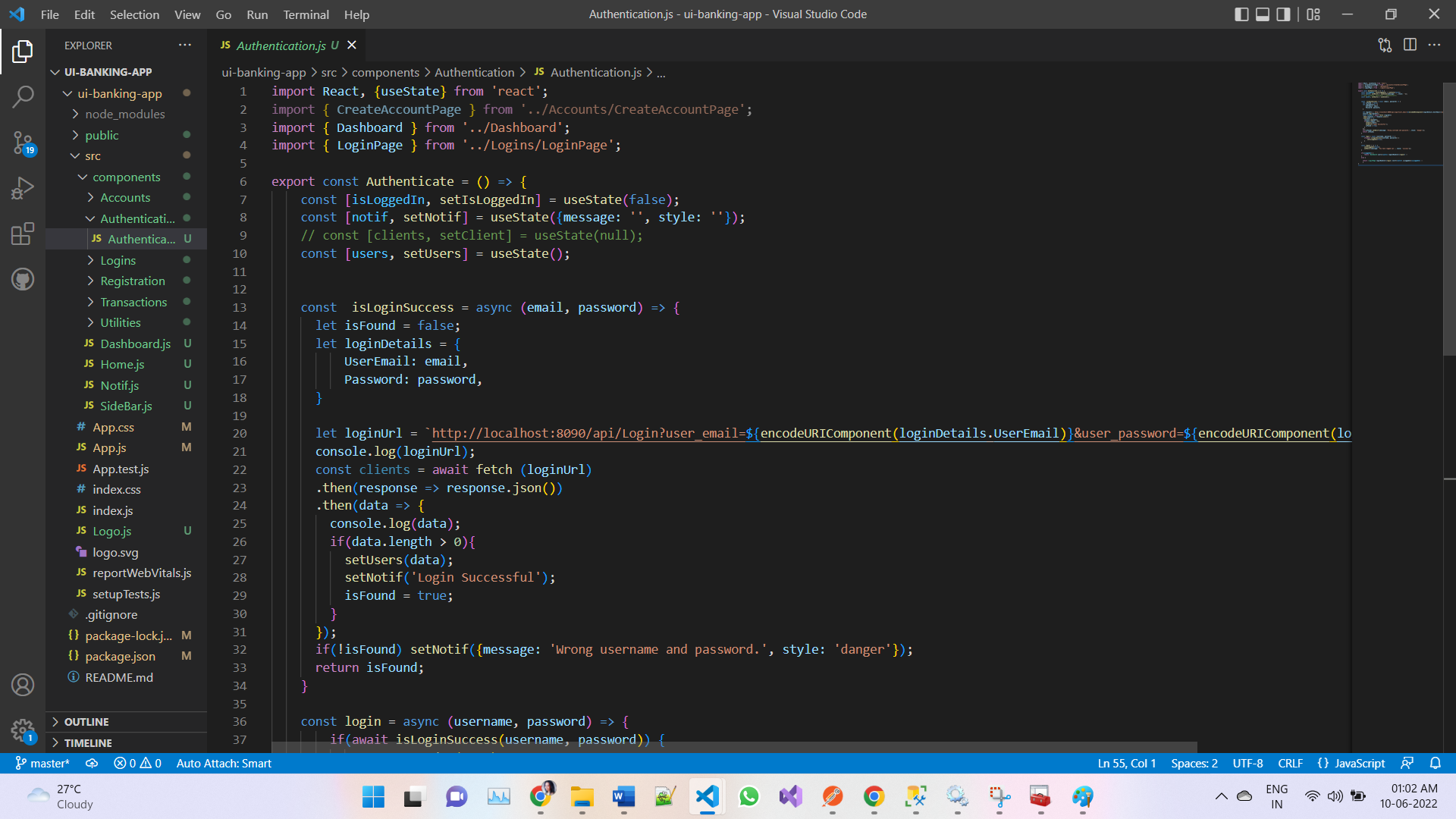
For Example: Below is a POST Call – with Request Body



Once the API is hit and operation is successful – **It returns 201 Created HTTP response**

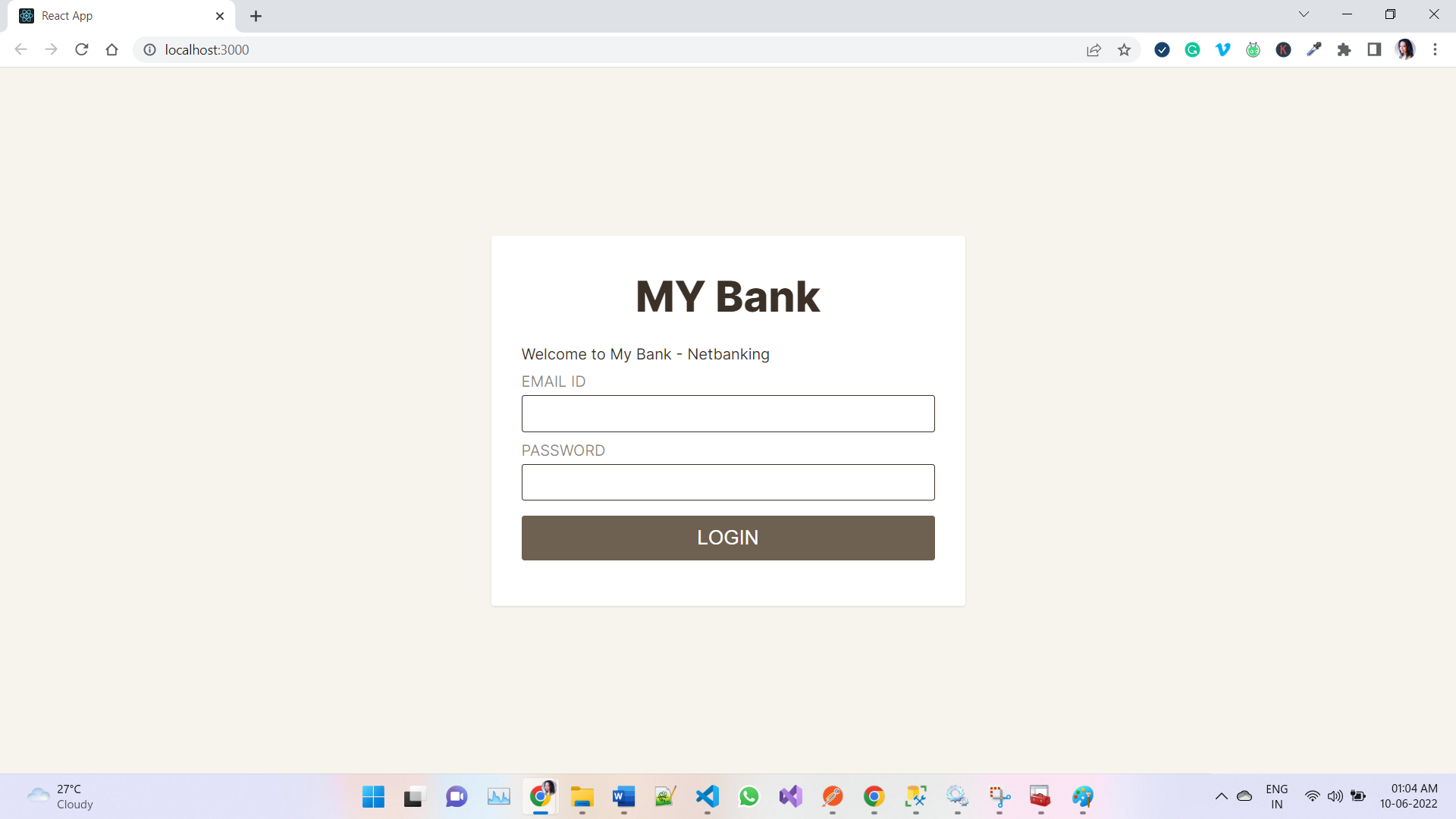
# FRONT END - REACTJS SET-UP

**PS: For Front End CSS and Styling– I have taken help from a source as I wasn’t finding time to work full-fledge, but the implementation is completely done by me.**

Once you unzip “**ui-banking-app**” you need to install the npm packages

Once the code is downloaded for **ui-banking-app**

* Run “**npm install**” to install the node modules according to package.json
* Next you need to go to the directory level where node\_modules are present in order to run the application – **cd .\ui-banking-app**
* Next you need to start the React.js Application by running **“npm start”** in the terminal
* You will see a screen like this



**Please insert below Login just for first time – (I did not get time to work on Registration)**

**EMAIL ID:** [**apurvatare05@gmail.com**](mailto:apurvatare05@gmail.com)

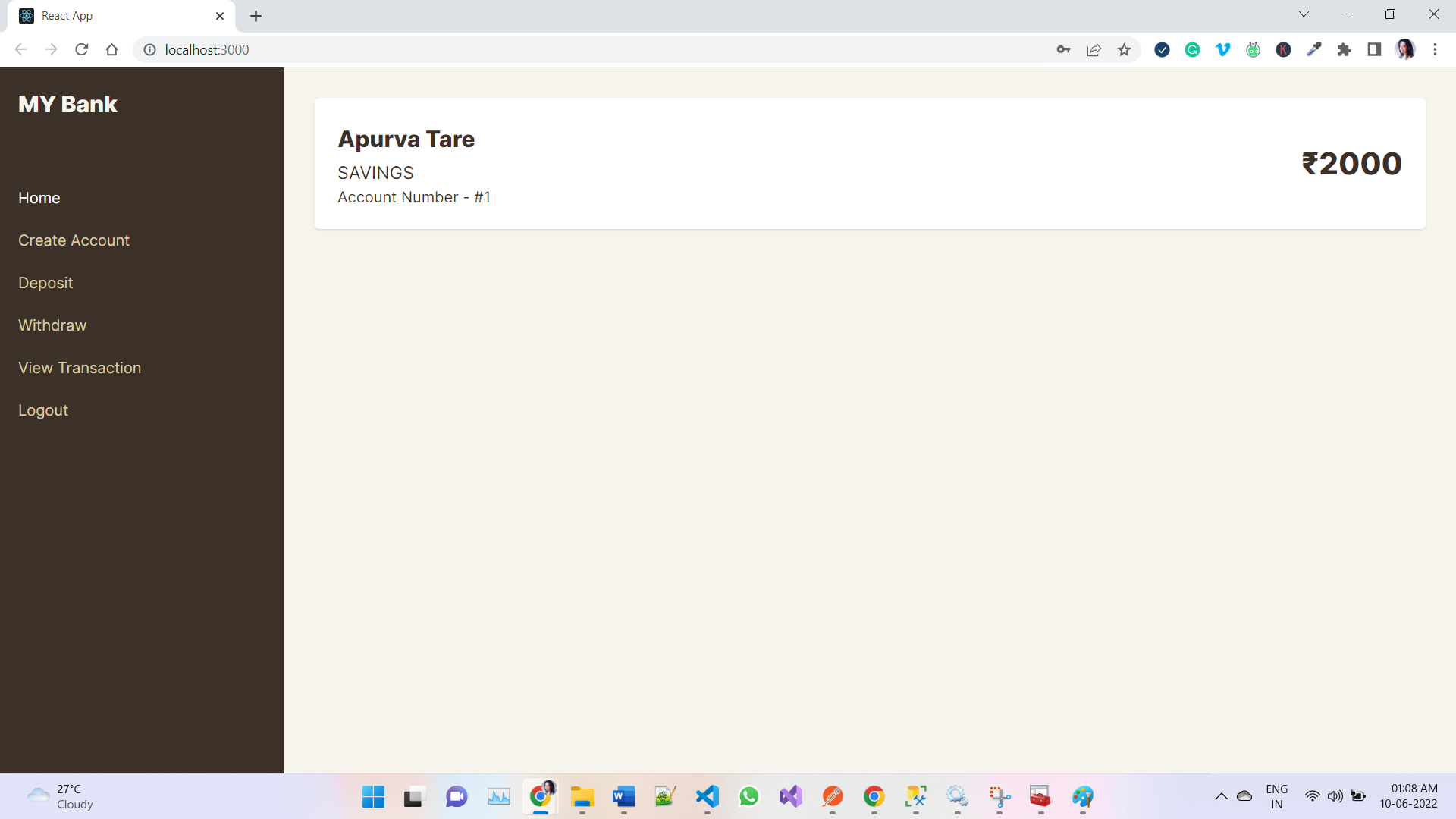
**Password: 123**

Once You Login

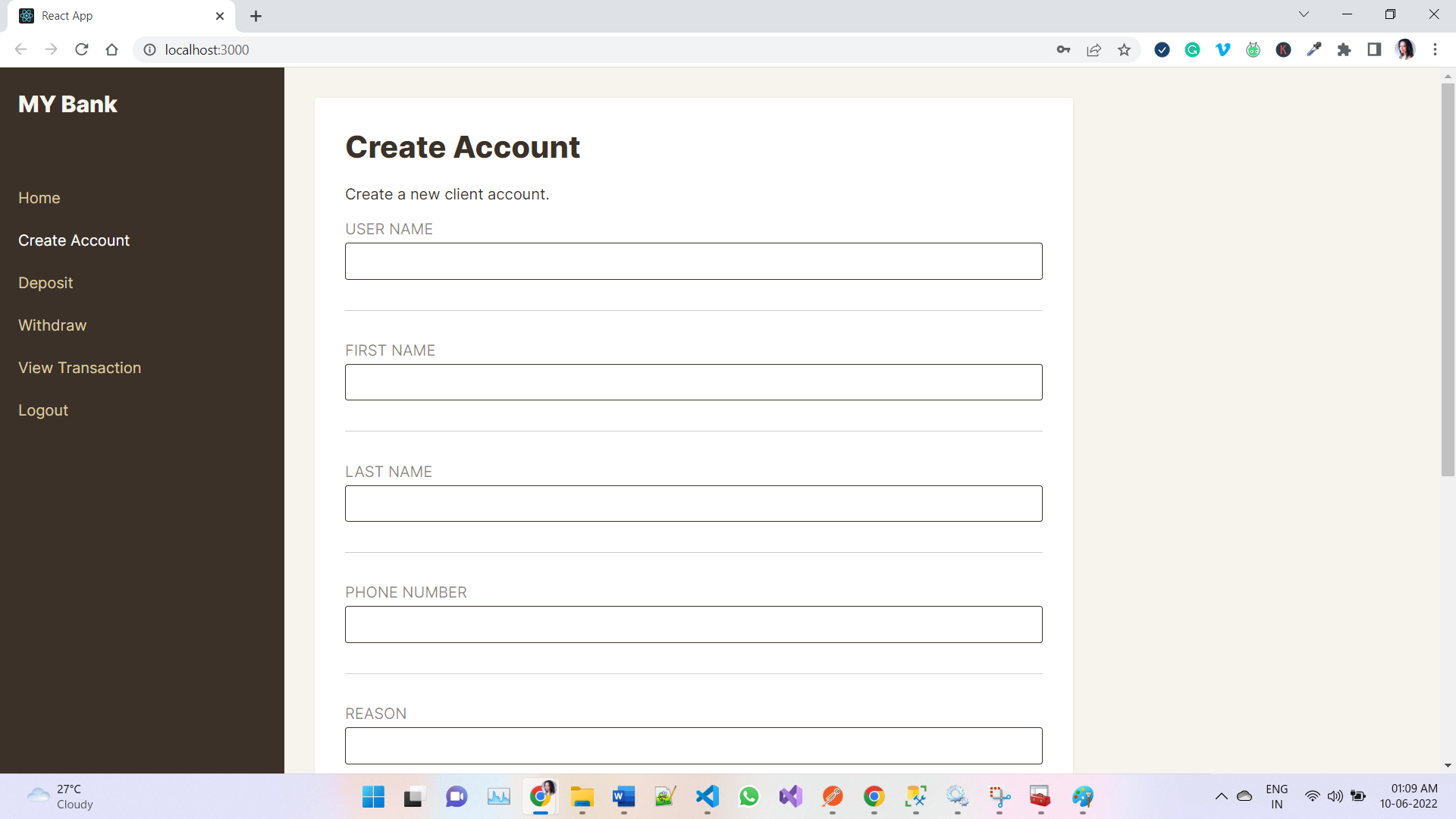
You will see a sidebar which has following Pages

1. Home Page – To Display Accounts
2. Create Account Page – Where you can create Accounts
3. Deposit – To make deposits to your selected account
4. Withdraw – To make withdrawals from your selected Accounts
5. View Transactions – To View Transactions
6. Logout

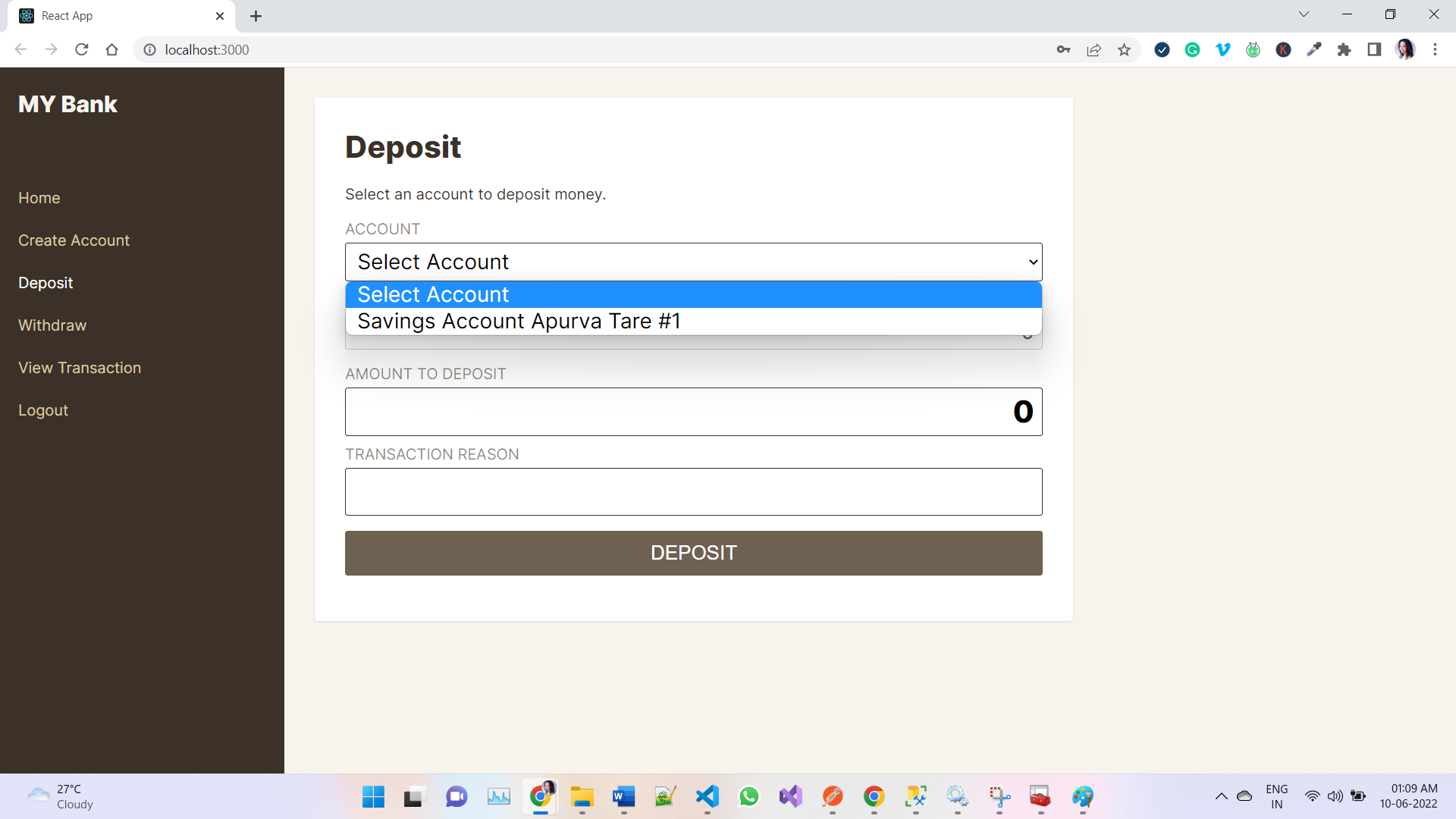
## Home Page

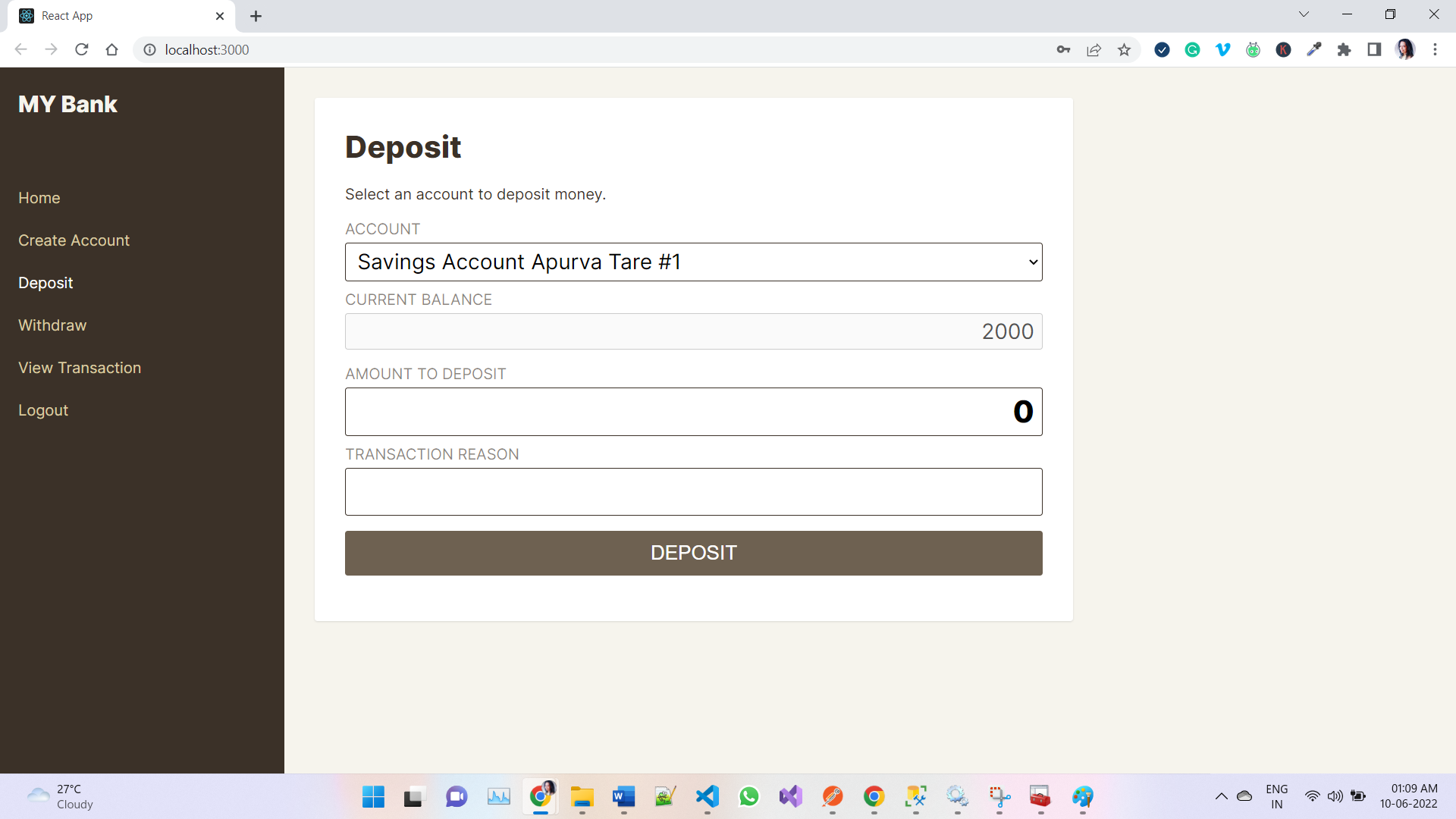


## Create Account Page

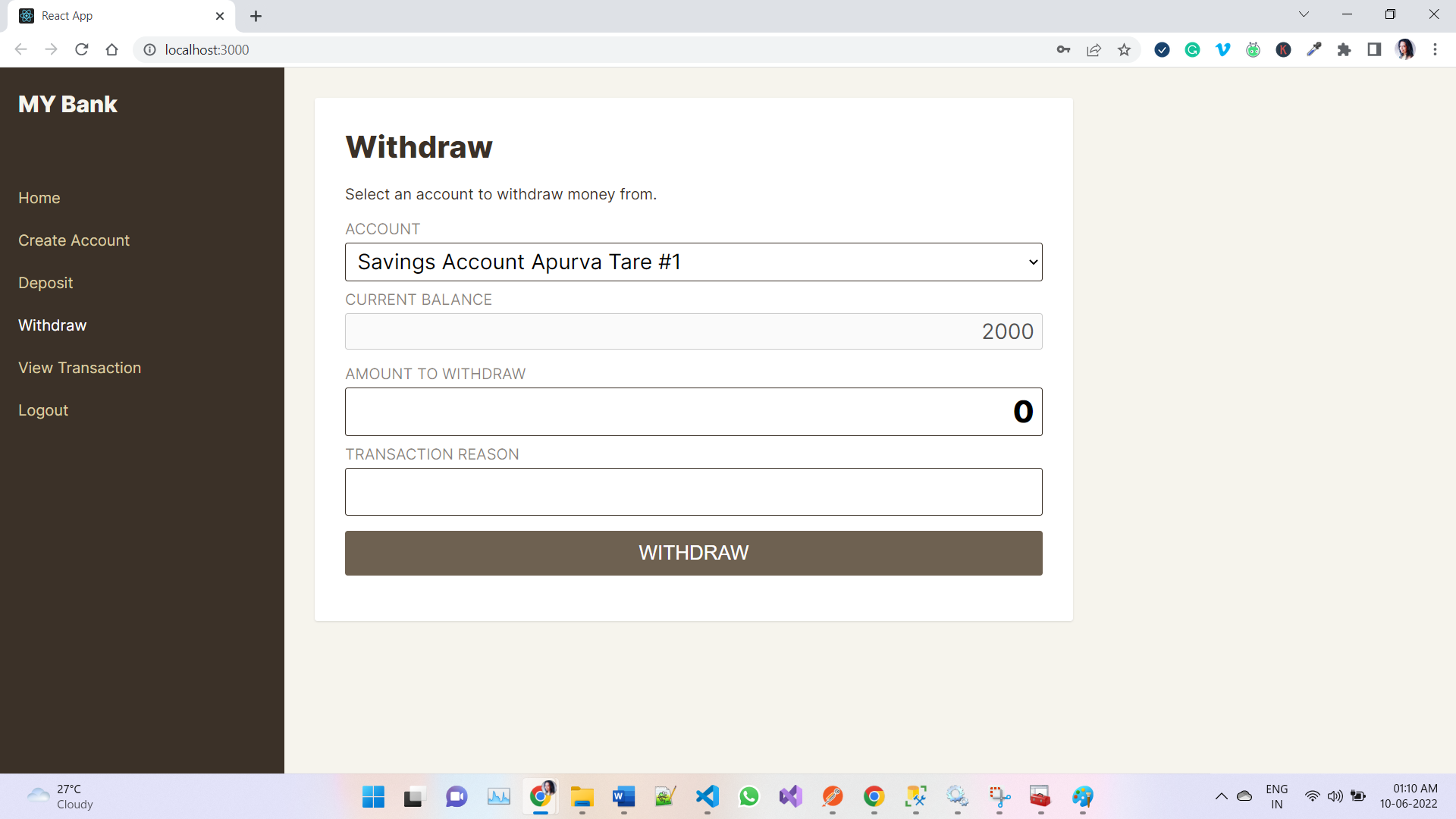


## Deposit Page

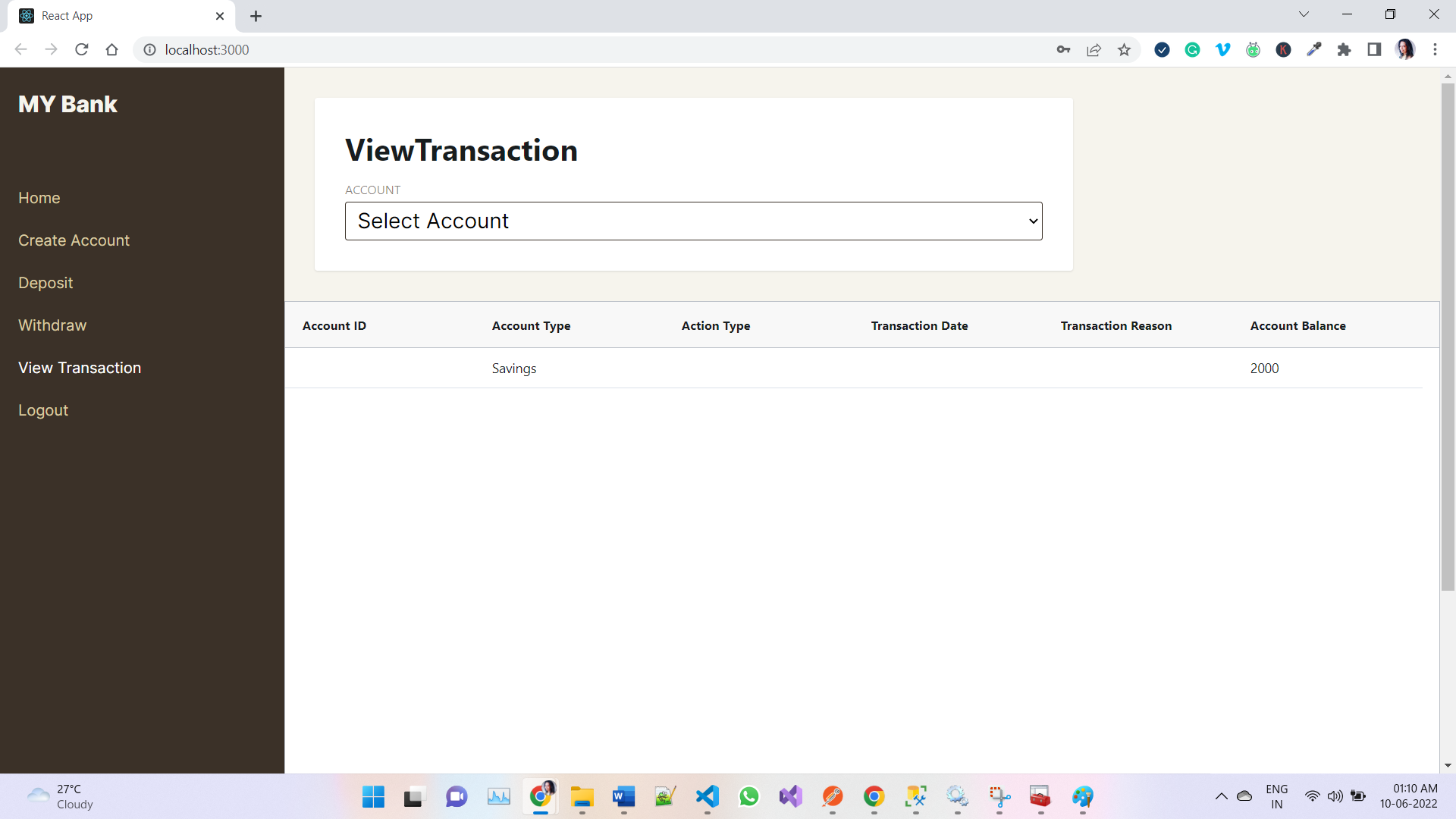


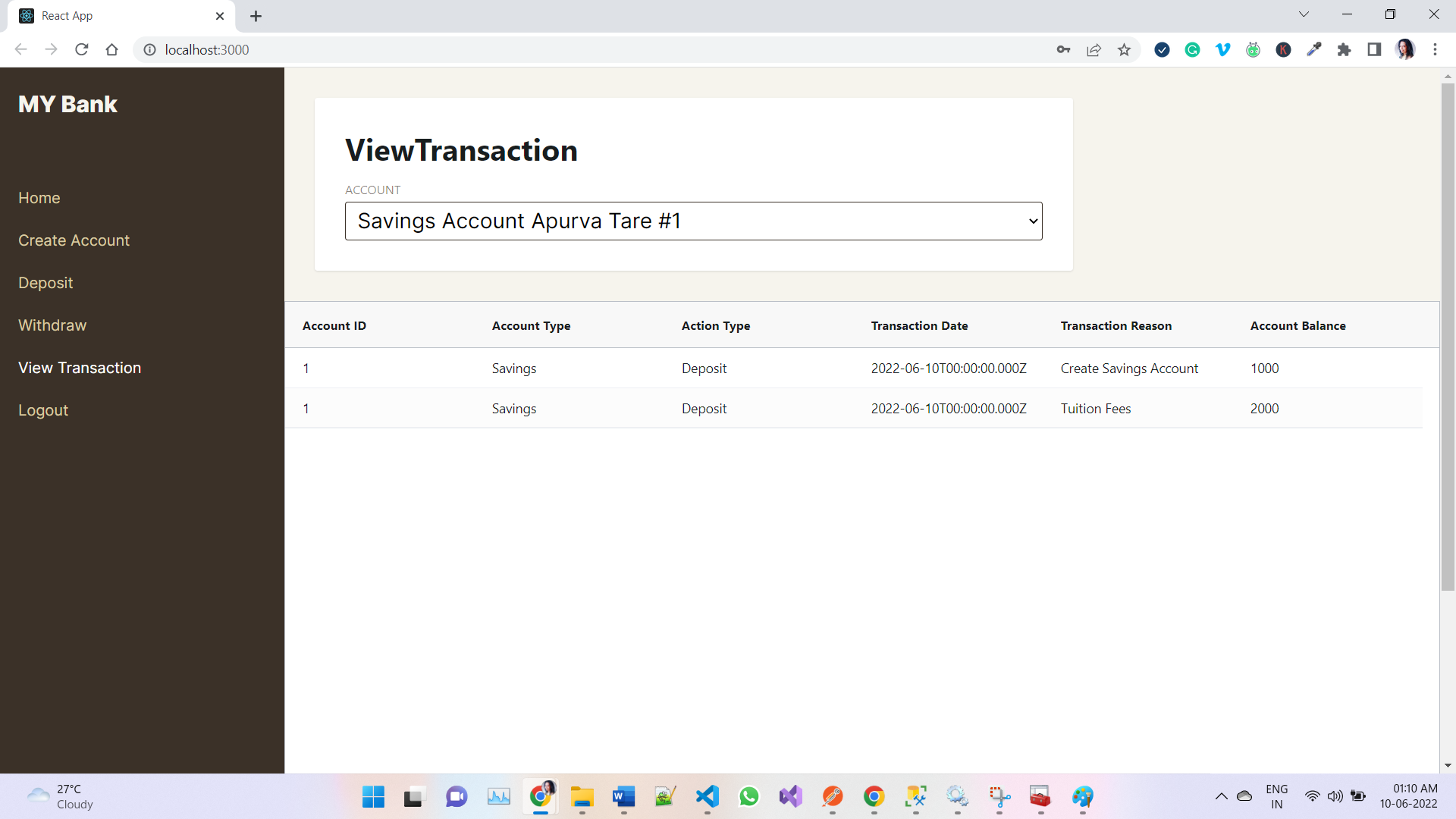


## Withdraw Page

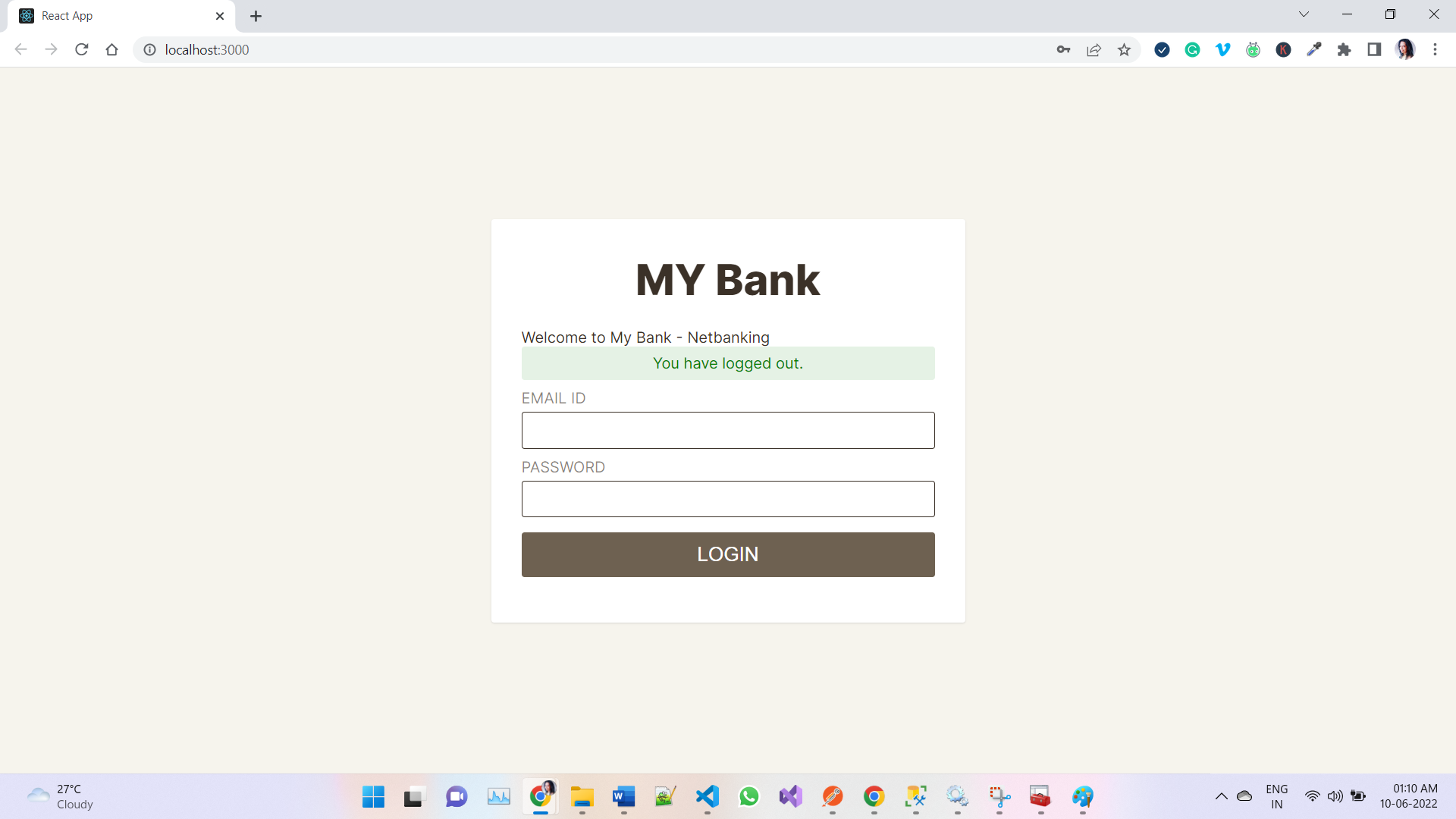


## View Transaction Page



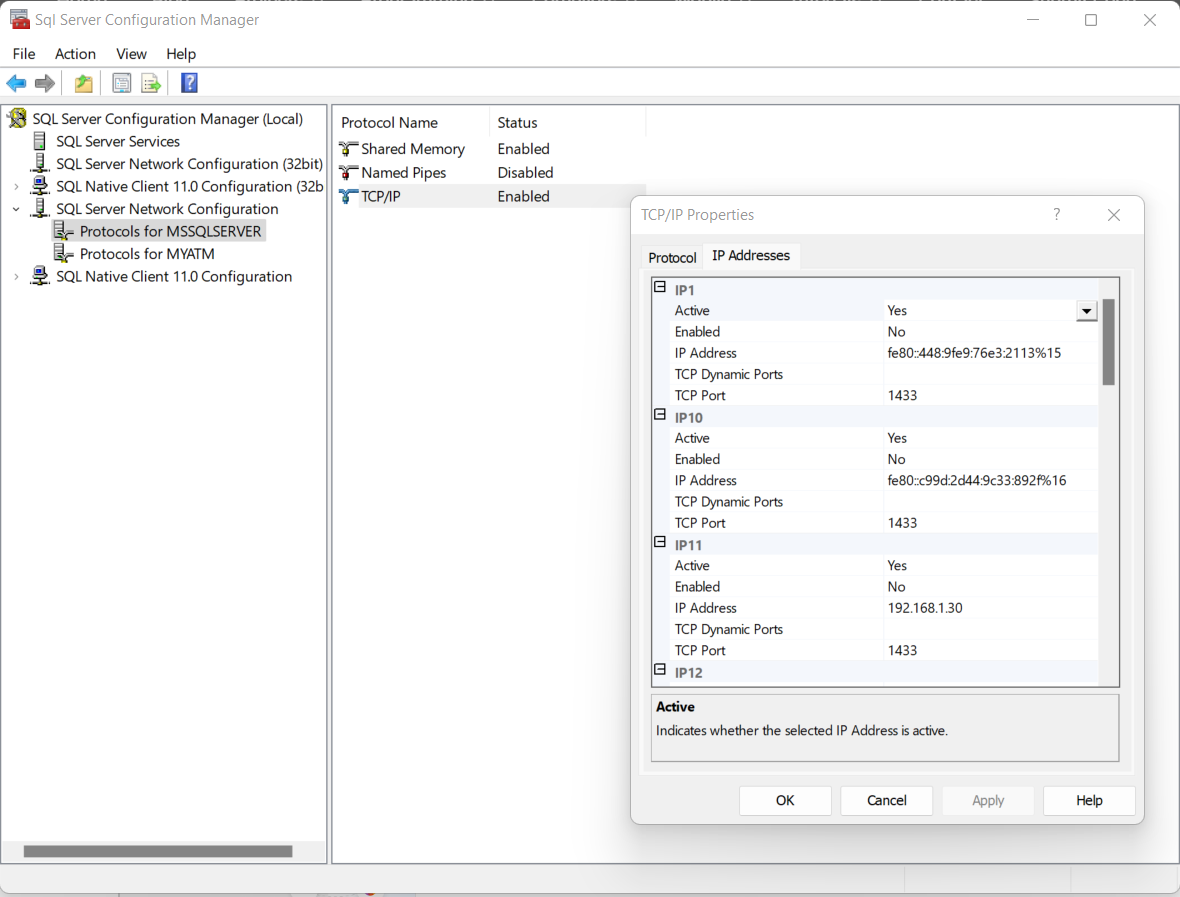


## Logout Page



# ISSUES FACED DURING THE ENTIRE IMPLEMENTATIONS

1. While Connecting Node.js to SQL Server – I wasn’t able to connect – there were a lot of errors like below
   1. ConnectionError: Failed to connect to servername\instantname in 15000ms – Issue in the dbConfig File
   2. Failed to lookup instance on DESKTOP-OMPTOBAMYATM - getaddrinfo ENOTFOUND DESKTOP-OMPTOBAMYATM – SQL Server Browser Instance needs to be running AND TCP/IP connection must be enabled and the port shouldn't be included within the server string, it must go inside the configuration as a numeric value.



* 1. Error: self-signed certificate in certificate chain – Added “encrypt: false” in dbConfig {options}
  2. SQLServerException: Login failed for user ‘ ‘ – Added driver: 'msnodesqlv8',

1. While Connecting ReactJS to NodeJS
   1. Failed to execute 'json' on 'Response': body stream already read - the real problem is that you only can consume Response.json() once, if you are consuming it more than once, the error will happen.

# REFERENCE MATERIALS USED to resolve errors

1. <https://www.mssqltips.com/sqlservertip/2340/resolving-could-not-open-a-connection-to-sql-server-errors/>
2. <https://gsferreira.com/archive/2014/12/overcome-the-depth_zero_self_signed_cert-on-nodejs/#:~:text=If%20you%20are%20making%20requests%20to%20a%20server,if%20you%20are%20working%20in%20a%20test%20environment>.
3. <https://stackoverflow.com/questions/53511974/javascript-fetch-failed-to-execute-json-on-response-body-stream-is-locked>
4. <https://stackoverflow.com/questions/61484005/timeout-error-while-connecting-to-sql-server-connection-error-failed-to-connec>