Spring Boot Security

Introduction to Security Concepts:

Security of web application:

- It is process of enabling **Authentication + Authorization** on the web application.

Authentication:

- Checking the **IDENTITY USER** by using **usernames** and **passwords**, **thumb impressions, iris**, **digital signatures**, **OTP,** etc.

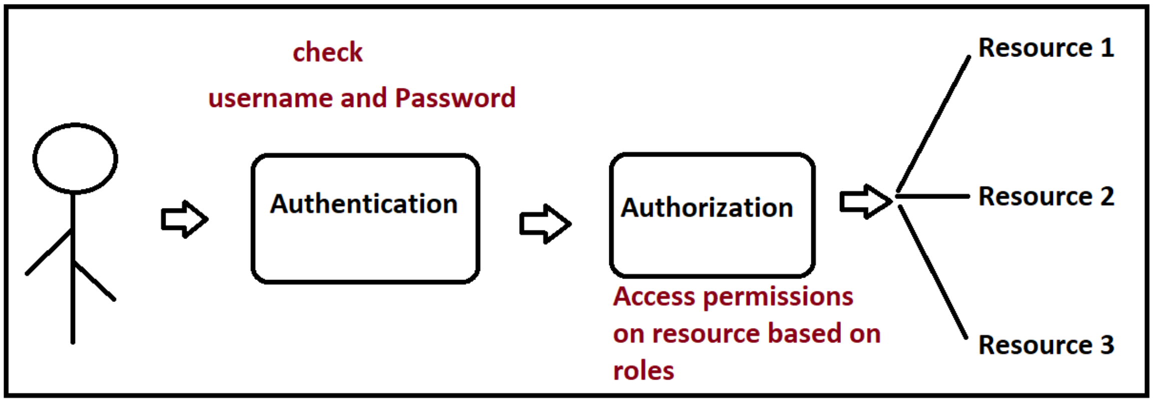
Authorization:

- Checking the access permissions of the authenticated users on resources of the application.
- Here the **roles** of the user will be verified before allowing the user to access resources of the application.
- **Roles** are nothing but designations given to the users.
- Based on the roles of the users, the access permissions on resources will be decided.
  Eg:

  All Customers and Employees of the bank must be authenticated to use **XYZ Bank Application**.
  The users having **Customer Role** will get less permission on the resources.
  The users having **Employee Role** will get more access permission on the resources.
- It is always recommended to enable **authorization** of accessing the resources based on **Roles of the users**, but not based on the username.
- During the authentication process, get the Roles of the users and use those roles for authorization.

Realm:

- It is a small database s/w or repository where **usernames, passwords and roles are managed.**



Authentication Providers:

- It is small realm where **usernames and passwords, and roles** are managed and will be used during Authentication and Authorization.

- The following are the different Authentication Providers.
    1. Properties file
    2. XML file
    3. JSON file
    4. DB s/w
    5. LDAP Server (Lightweight Directory Access Protocol)
    6. In-Memory DB

Authentication and Authorization Manager:

- It is the component which verifies the given **username** and **password** to perform Authentication and give **401 error code** if authentication fails.

- It also collects the roles of authenticated users and performs Authorization activities on access resources and gives **403 error** if authorization fails.

- **Authentication and Authorization Manager** are provided by **Spring Security or Spring Boot Security**.

Authorization Levels in Spring Security:

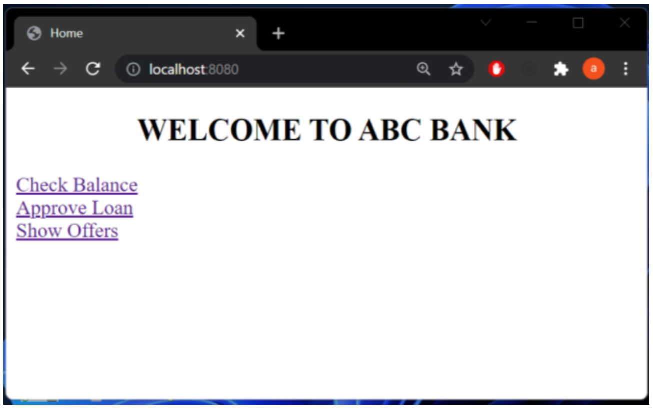
1. **permitAll():**
   No Authentication + No Authorization (No Role Checking)
   Eg:
   Home page, About Us page, Contact Us page, etc

2. **authenticate():**
   Only Authentication on the given request url resource (controller) and no Authorization (no role checking)
   Eg:
   Main Menu page, Inbox page, Compose page, etc.

3. **hasRole():**
   Authentication + Authorization (Role Checking ("USER"))
   Eg:
   Checking Balance page, Transfer Money page, Withdraw or Deposit money page, etc

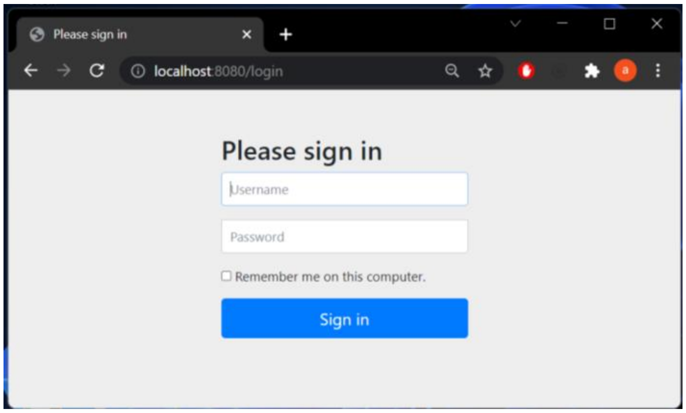
4. **hasAnyRole():**
   Authentication + Authorization (Any one Role should be there in the list of given roles ("USER", "MANAGER")
   Eg:
   Checking Balance page, Transfer Money page, Withdraw or Deposit money page, etc

USE CASE:



WELCOME TO ABC BANK

Check Balance
Approve Loan
Show Offers



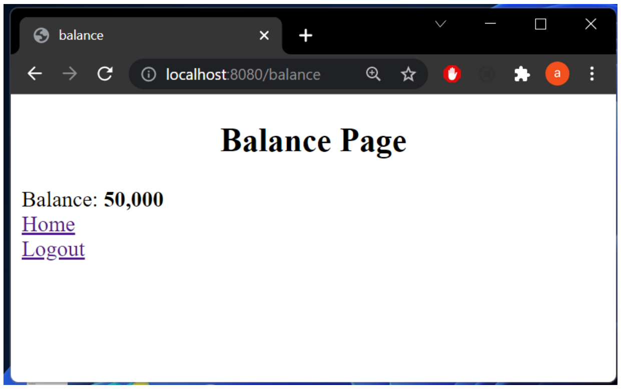
Please sign in

Username

Password

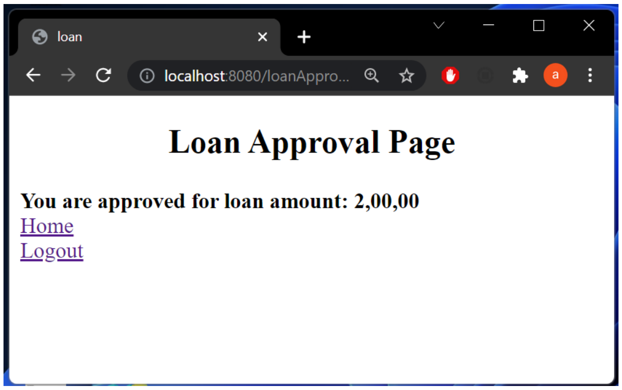
□ Remember me on this computer.

Sign in



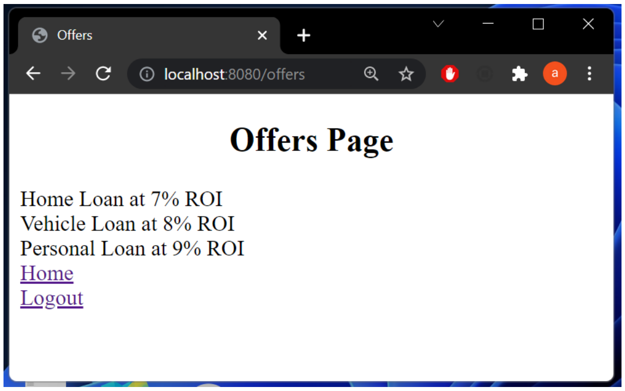
Balance Page

Balance: **50,000**
Home
Logout



Loan Approval Page

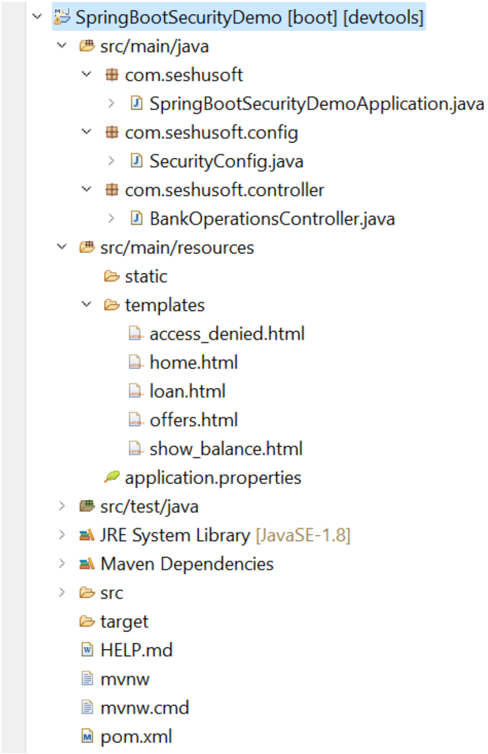
You are approved for loan amount: 2,00,00
Home
Logout



Offers Page

Home Loan at 7% ROI
Vehicle Loan at 8% ROI
Personal Loan at 9% ROI
Home
Logout

Create Spring Starter project
Dependencies: Spring dev tools, Spring Web, Spring Security, thymeleaf



```
SpringBootSecurityDemo [boot] [devtools]
  src/main/java
    com.seshusoft
      SpringBootSecurityDemoApplication.java
    com.seshusoft.config
      SecurityConfig.java
    com.seshusoft.controller
      BankOperationsController.java
  src/main/resources
    static
    templates
      access_denied.html
      home.html
      loan.html
      offers.html
      show_balance.html
    application.properties
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

home.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Home</title>
</head>
<body>
<h2 style="text-align: center;">WELCOME TO ABC BANK</h2>
    <a href="balance">Check Balance</a>
    <br>
    <a href="loanApprove">Approve Loan</a>
    <br>
    <a href="offers">Show Offers</a>
</body>
</html>
```

access_denied.html

```
<!DOCTYPE html>
<html>
<head>
    <title>access_denied</title>
</head>
<body>
<h2 style="text-align: center;">Access Denied Page</h2>
<a href="./">Home</a>
</body>
</html>
```

loan.html

```
<!DOCTYPE html>
<html>
<head>
    <title>loan</title>
</head>
<body>
<h2 style="text-align: center;">Loan Approval Page</h2>
    <b>You are approved for loan amount: 2,00,00</b>
    <br>
    <a href="./">Home</a>
    <br>
    <a href="logout">Logout</a>
</body>
</html>
```

offers.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Offers</title>
</head>
<body>
<h2 style="text-align: center;">Offers Page</h2>
    Home Loan at 7% ROI
    <br>
    Vehicle Loan at 8% ROI
    <br>
    Personal Loan at 9% ROI
    <br>
    <a href="./">Home</a>
     <br>
    <a href="logout">Logout</a>
</body>
</html>
```

show_balance.html

```
<!DOCTYPE html>
<html>
<head>
    <title>balance</title>
</head>
<body>
<h2 style="text-align: center;">Balance Page</h2>
    Balance: <b>50,000</b>
    <br>
    <a href="./">Home</a>
    <br>
    <a href="logout">Logout</a>
</body>
</html>
```

BankOperationsController.java

```java
package com.seshusoft.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class BankOperationsController {
	@GetMapping("/")
	public String showHome() {
		return "home";
	}

	@GetMapping("/offers")
	public String showOffers() {
		return "offers";
	}

	@GetMapping("/balance")
	public String showBalance() {
		return "show_balance";
	}

	@GetMapping("/loanApprove")
	public String approveLoan() {
		return "loan";
	}

	@GetMapping("/denied")
	public String accessDenied() {
		return "access_denied";
	}
}
```

SecurityConfig.java

```
package com.seshusoft.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import 
org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuil
der;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import 
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter{

       @Bean
       public PasswordEncoder encoder() {
           return new BCryptPasswordEncoder();
       }

       @Override
       public void configure(AuthenticationManagerBuilder auth) throws Exception {
             auth.inMemoryAuthentication()
                    .withUser("adi")
                    .password(encoder().encode("adi123"))
                    .roles("CUSTOMER");
             auth.inMemoryAuthentication()
                    .withUser("seshu")
                    .password(encoder().encode("seshu123"))
                    .roles("MANAGER");
       }

       @Override
       public void configure(HttpSecurity http) throws Exception {
             http.authorizeRequests()
             //No authentication and No authorization
             .antMatchers("/").permitAll()

             //Only Athentication
             .antMatchers("/offers").authenticated()

             //authentication + authorization for "CUSTOMER", "MANAGER" role users
             .antMatchers("/balance").hasAnyRole("CUSTOMER","MANAGER")

             //authentication + authorization for "MANAGER" role user
             .antMatchers("/loanApprove").hasRole("MANAGER")

             //remaining all request urls must be authenticated
             .anyRequest().authenticated()

             //authentication mode
             .and().formLogin()
```

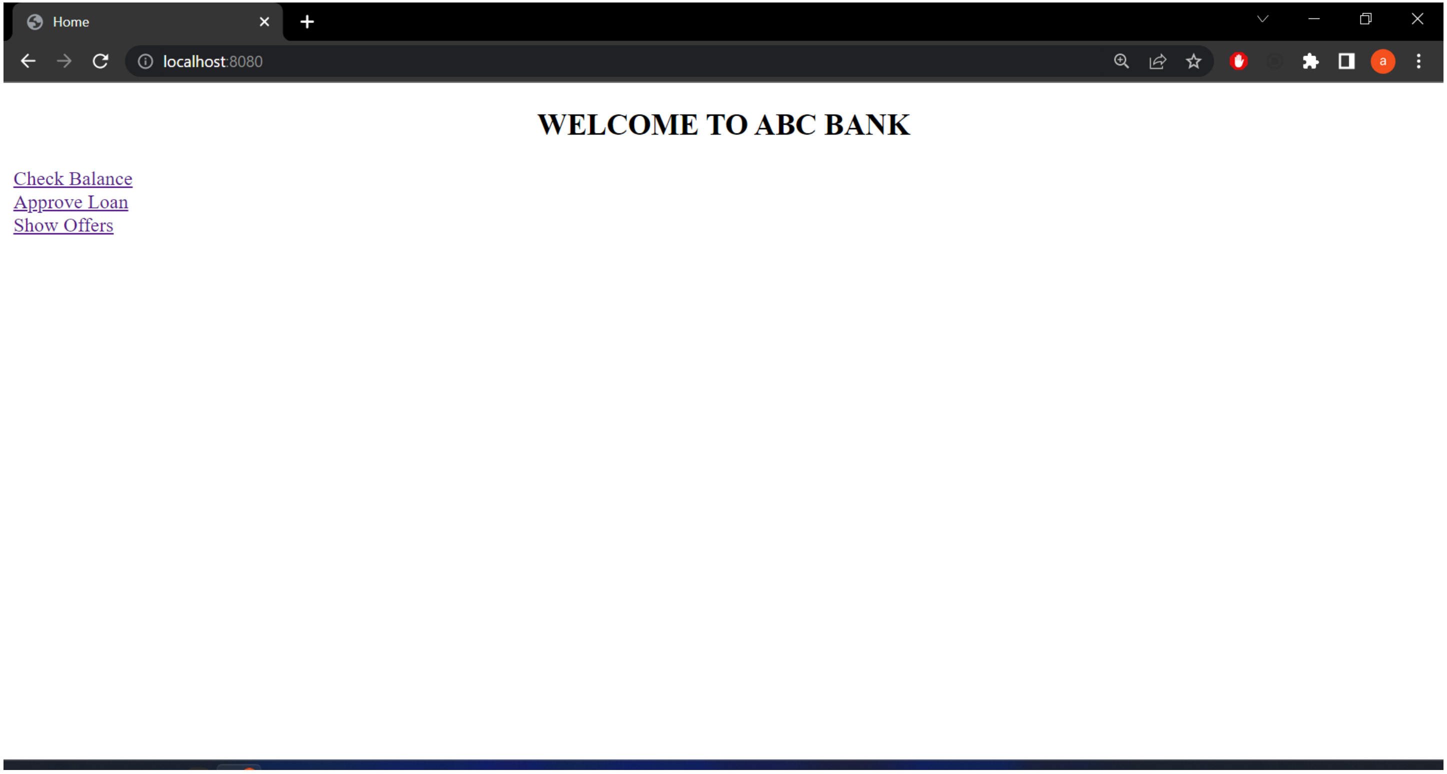
```
                .and().rememberMe()

                .and().logout().permitAll()

                //error handling
                .and().exceptionHandling().accessDeniedPage("/denied");
        }
}
```

Run Starter class

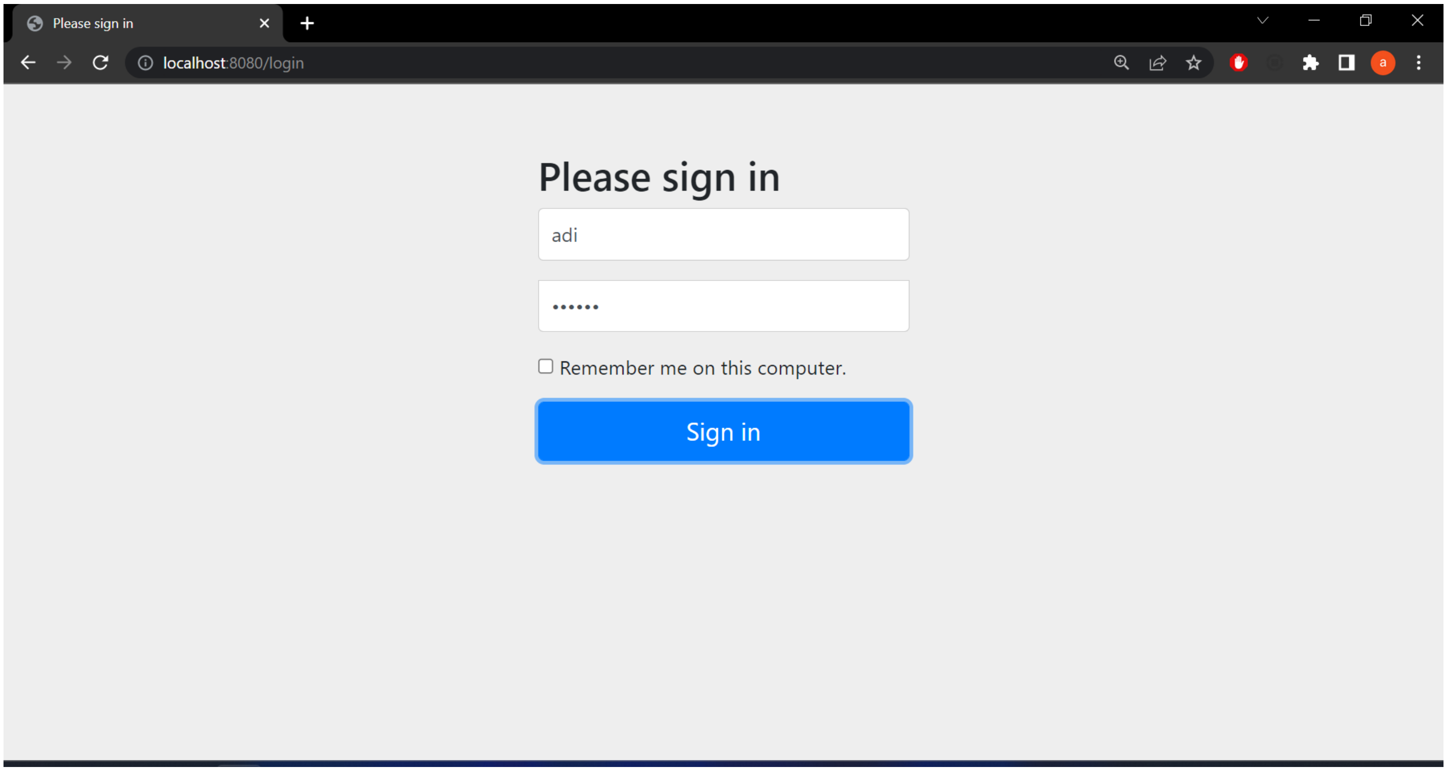
http://localhost:8080/



WELCOME TO ABC BANK

Check Balance
Approve Loan
Show Offers

Click on Check Balance link
Provide credentials
Username : adi
Password: adi123

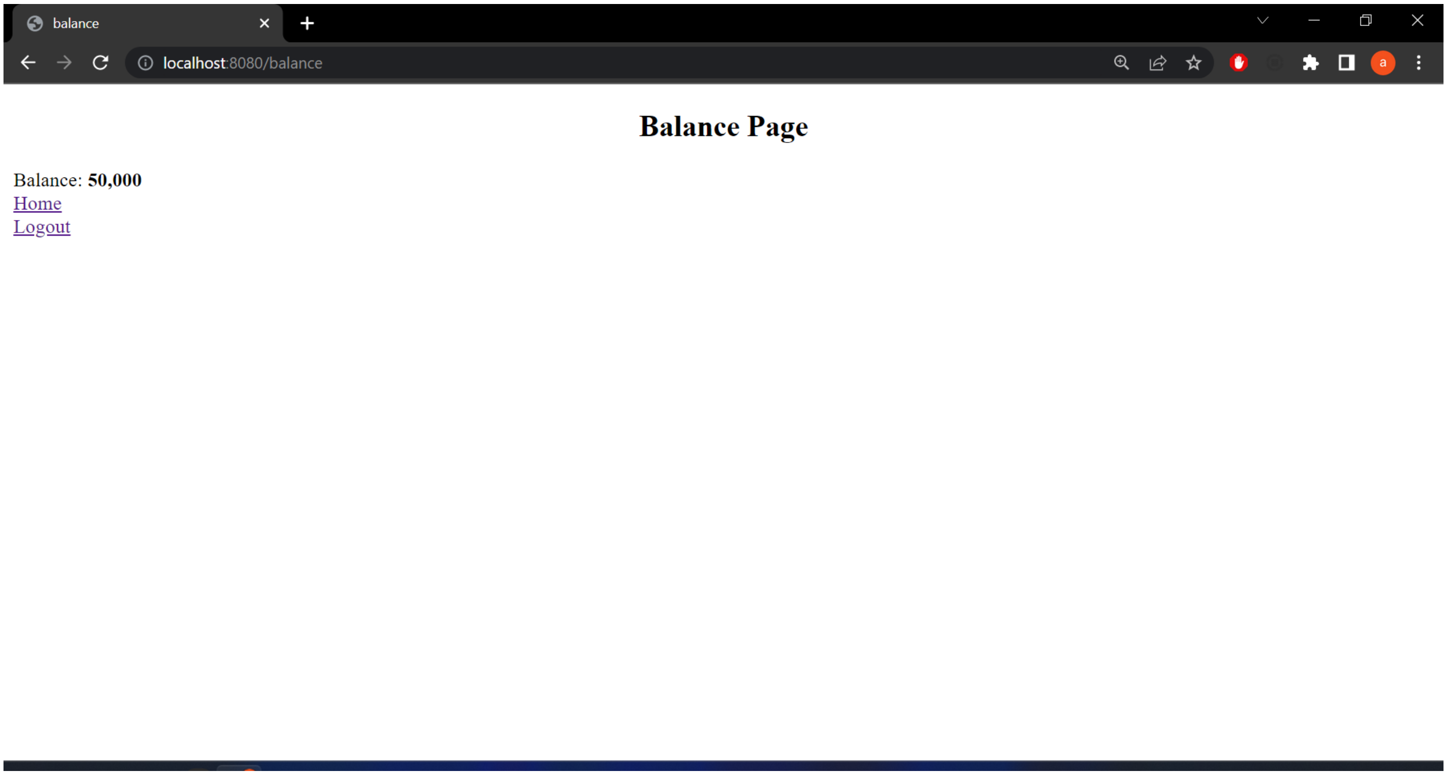


Please sign in

adi

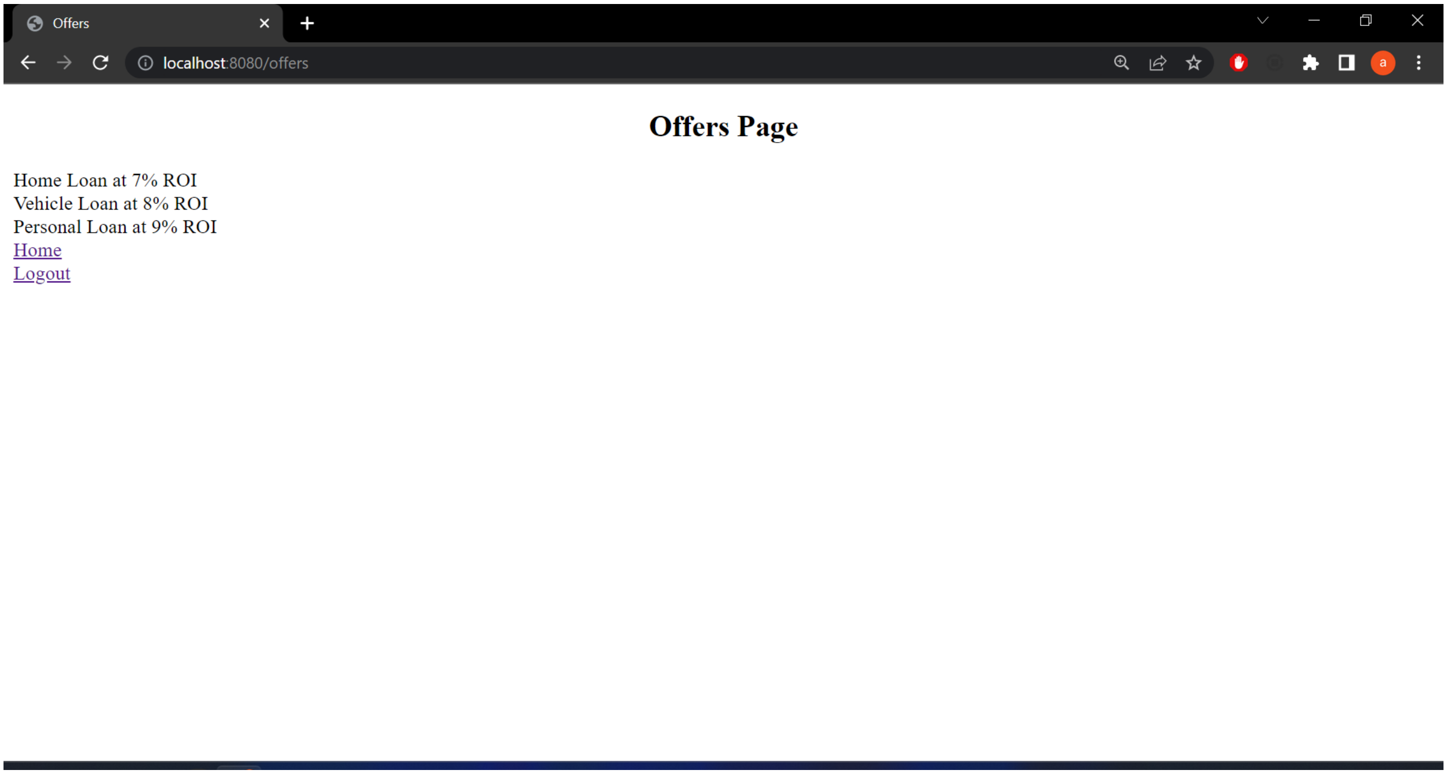
••••••

□ Remember me on this computer.



Balance Page

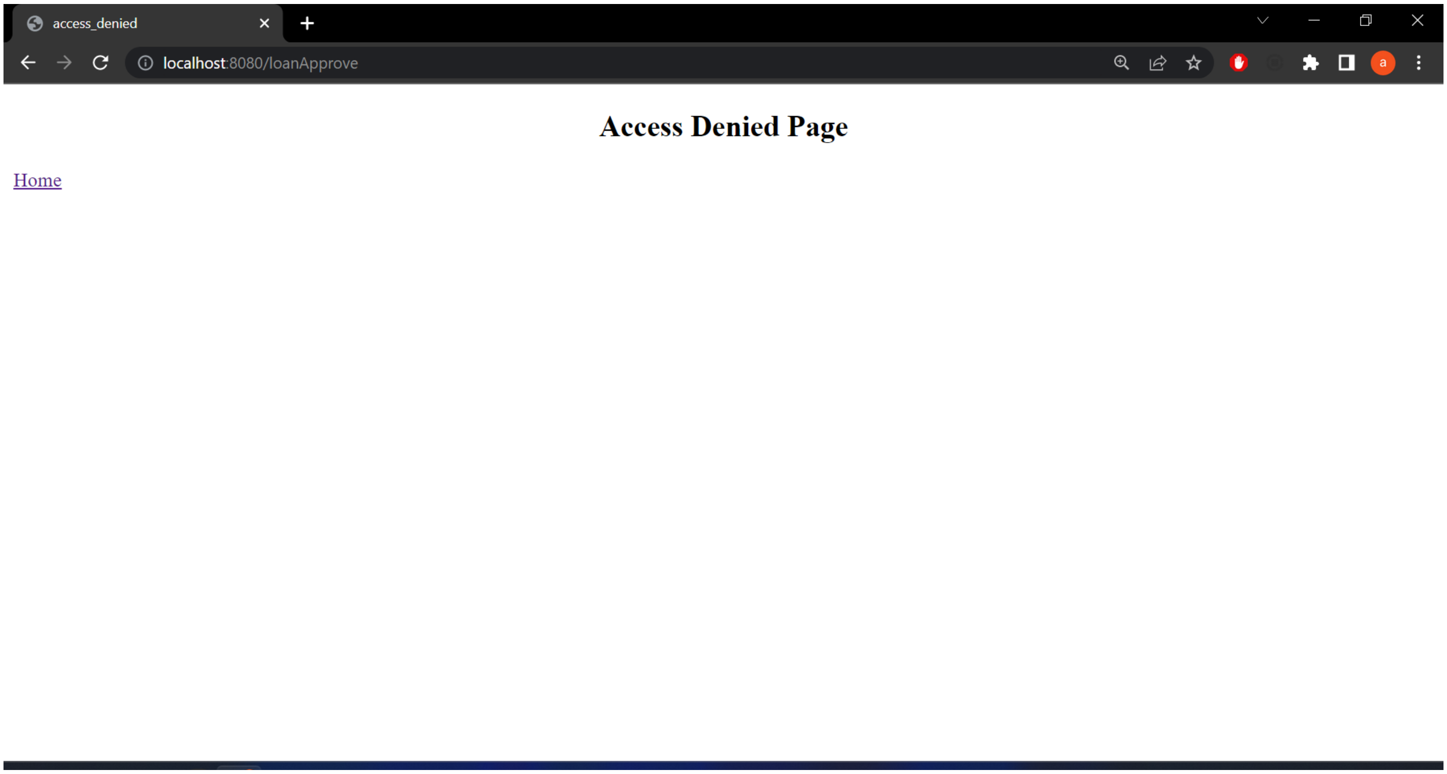
Balance: **50,000**
Home
Logout



Offers Page

Home Loan at 7% ROI
Vehicle Loan at 8% ROI
Personal Loan at 9% ROI
Home
Logout

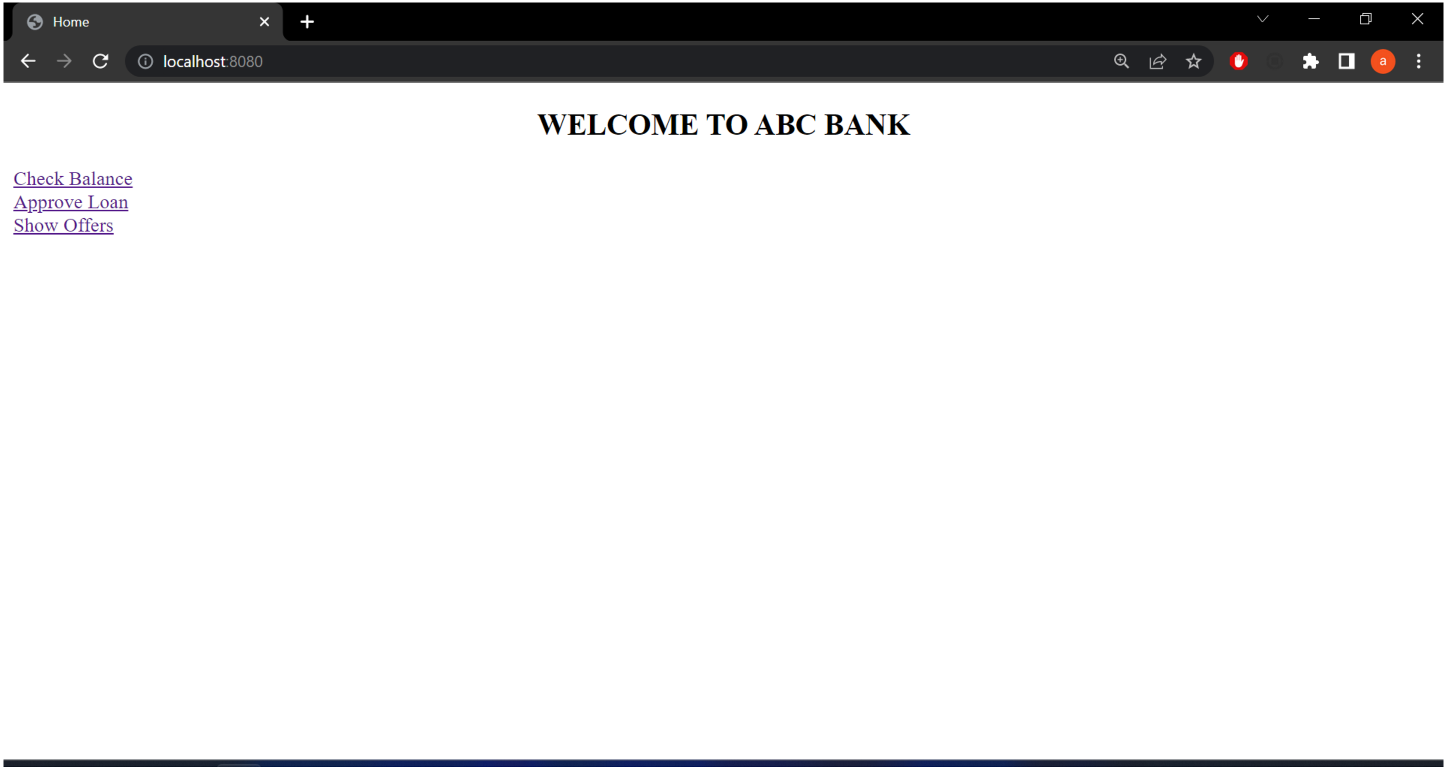
Click on Approve Loan link
As the user adi having only CUSTOMER role, not able to access Approve Loan link.



Access Denied Page

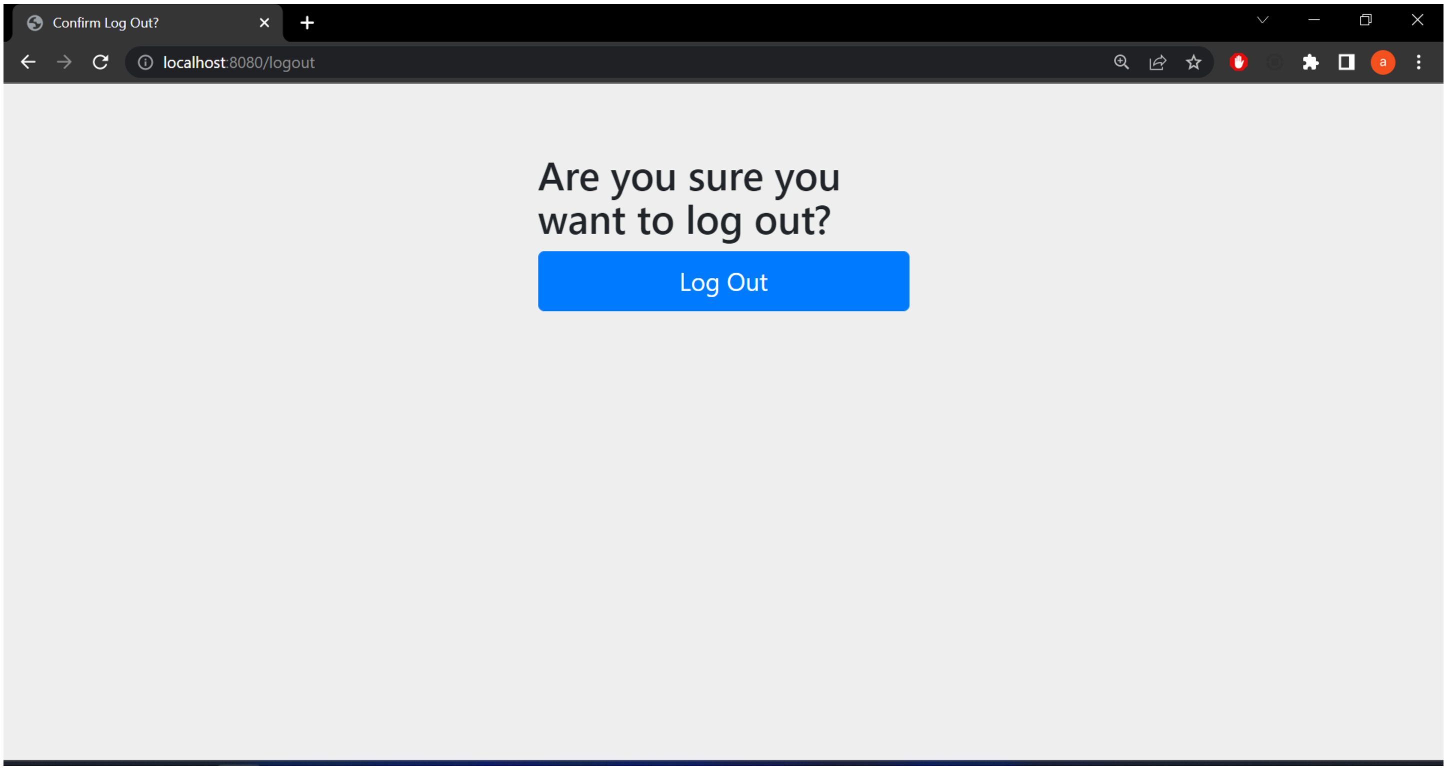
Home

Click Home and Logout.



WELCOME TO ABC BANK

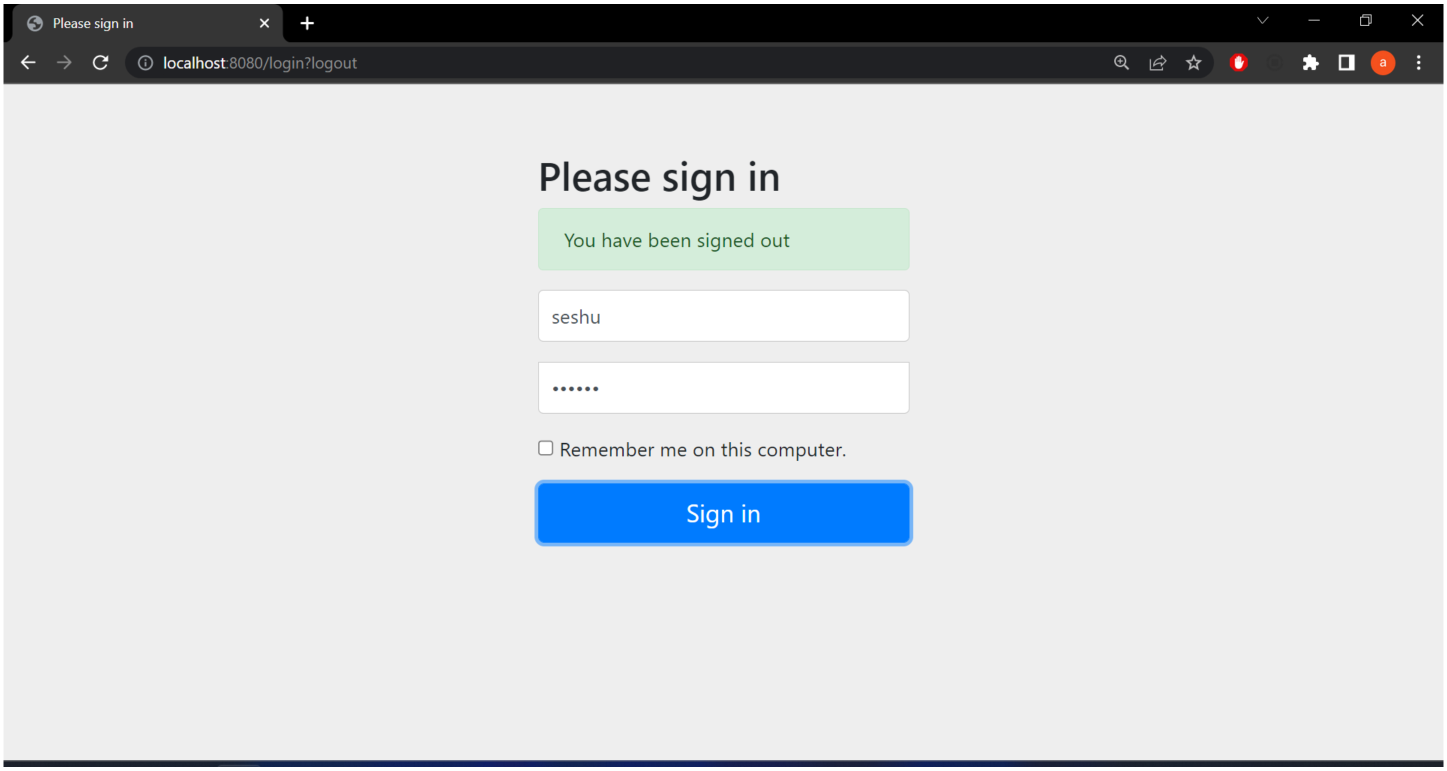
Check Balance
Approve Loan
Show Offers



Are you sure you want to log out?

Log Out

Once again login with credentials
Username: seshu
Password: seshu123



Please sign in

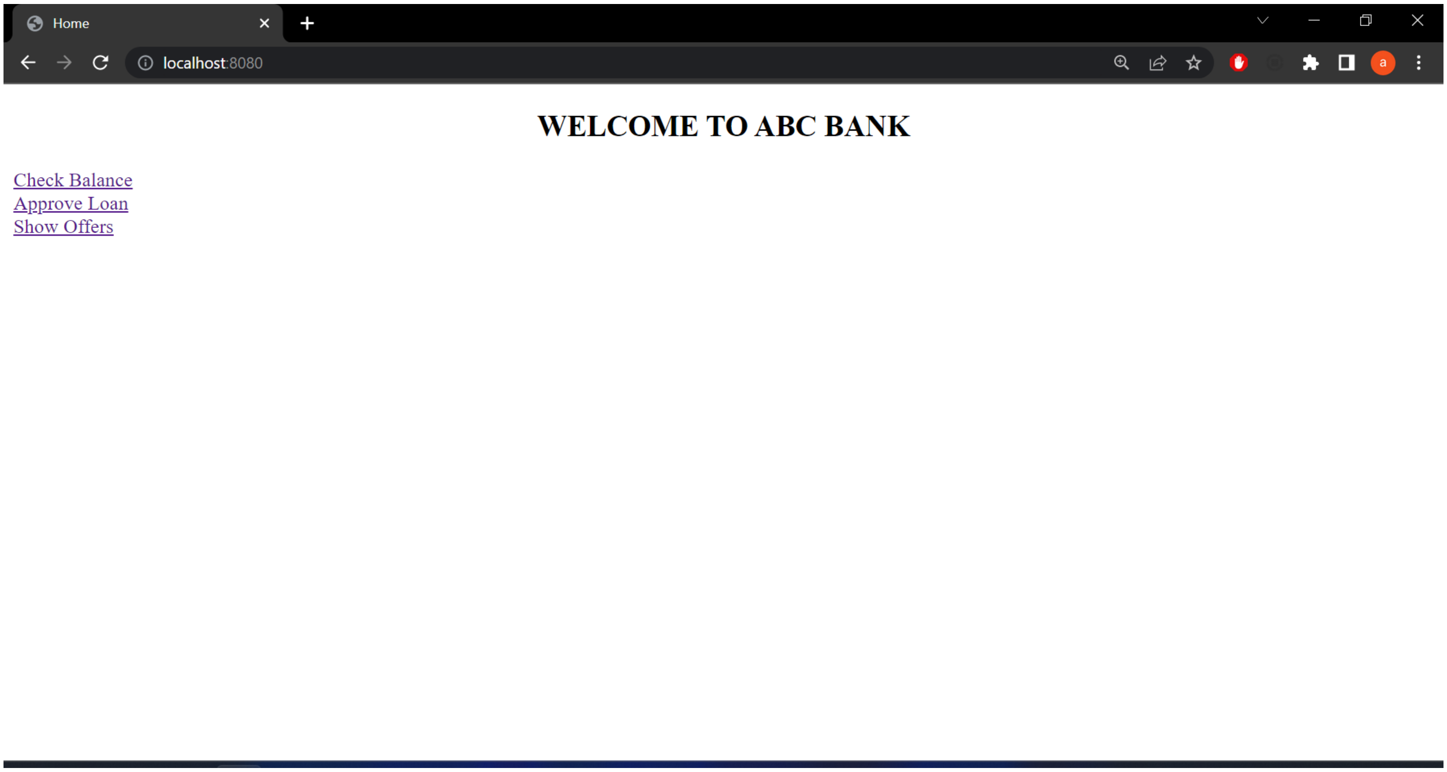
You have been signed out

seshu

••••••

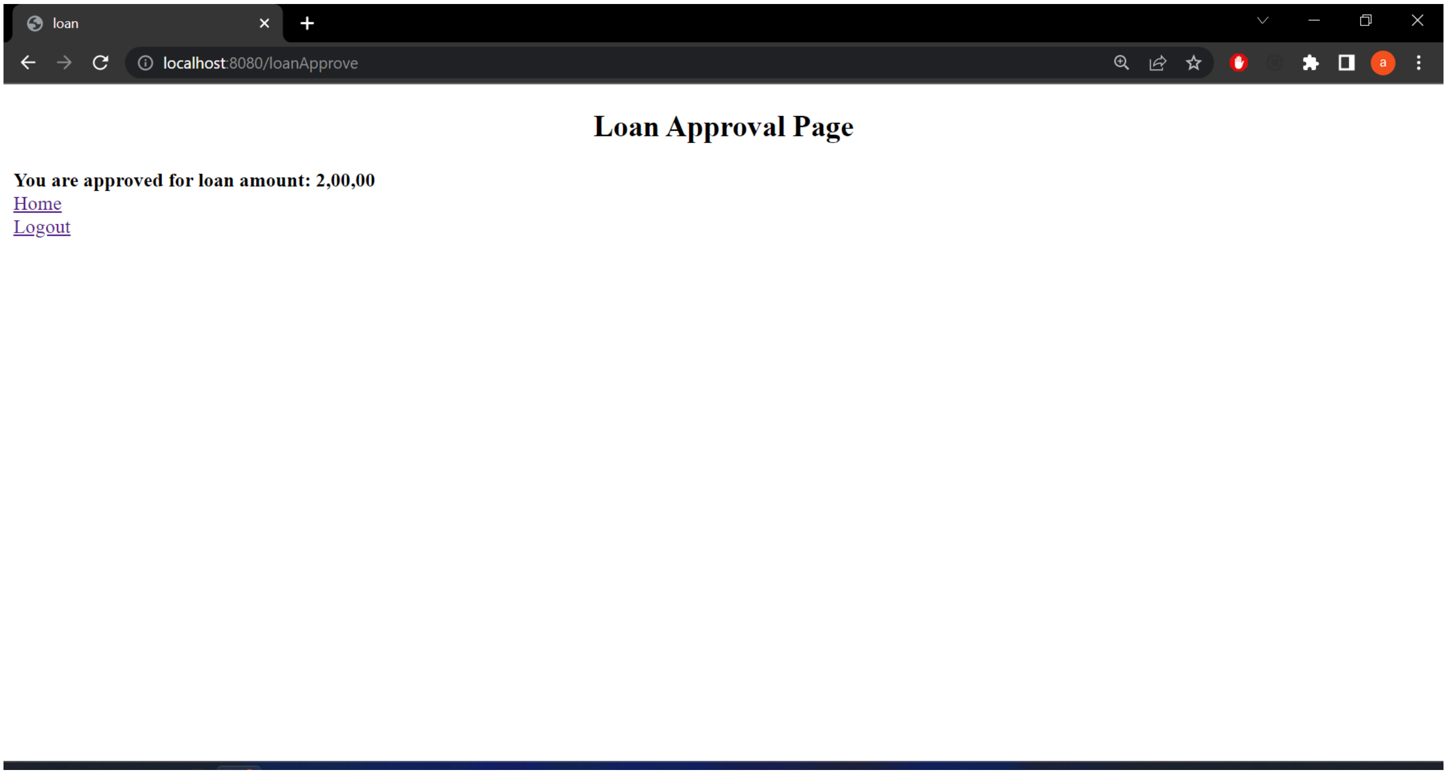
□ Remember me on this computer.

Now, click on Approve Loan



WELCOME TO ABC BANK

Check Balance
Approve Loan
Show Offers



Loan Approval Page

You are approved for loan amount: 2,00,00
Home
Logout