

Capstone Project - About

ICIN BANK

This document contains the following.

- Project and developer details
- Sprint planning and tasks achieved
- Core concepts used in the project
- Flowchart of the application
- Links to the GitHub repository
- Developer Detail
- Source Code
- Screen Shots of OutPut

Description

Create a dynamic and responsive Java online banking web application to deposit, withdraw, and transfer the money between the accounts.

Background of the problem statement:

ICIN is one of the top banking firms that accepts deposits from the public for the purpose of lending loans to the public. It also invests an amount in securities.

Recently, the business analysts noticed a drop in the number of customers of the bank. They found out that online banking systems of banks like AXIS and American Express are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer who can develop an online banking web application with a rich and user-friendly interface. You are hired as one of the Full Stack Java developers and have been asked to develop the web

application. The management team has provided you the requirements and their business model so that you can easily arrange different components of the application.

Features of the application:

1. Registration
2. Login
3. Account transactions
4. Transfers
5. Savings details
6. Profile settings
7. Requesting cheque books

Recommended technologies:

1. Database management: MySQL
2. Back-end logic: Java programming, SpringBoot framework
3. Front-end development: Angular 2, HTML/CSS,
4. Automation and testing technologies: Selenium and JUnit
5. DevOps and production technologies: Git, GitHub, Jenkins, Docker, and AWS

Project development guidelines:

- The project will be delivered within four sprints with every sprint delivering a minimal viable product.
- It is mandatory to perform proper sprint planning with user stories to develop all the components of the project.
- The learner can use any technology from the above-mentioned technologies for different layers of the project.
- The web application should be responsive and should fetch or send data dynamically without hardcoded values.
- The learner must maintain the version of the application over GitHub and every new change should be sent to the repository.
- The learner must implement a CI/CD pipeline using Jenkins.
- The learner should also deploy and host the application on an AWS EC2 instance.
- The learner should also implement automation testing before the application enters the CI/CD pipeline.
- The learner should use Git branching to separately perform the basic automation testing of application.
- The learner should make a rich front-end of the application, which is user- friendly and easy for the user to navigate through the application.
- There will be two portals in the application, namely the admin and user portal.

Admin Portal:

It deals with all the back-end data generation and product information. The admin user should be able to:

- Authorize the roles and guidelines for the user
- Grant access to the user regarding money transfer, deposits, and withdrawal
- Block the user account in case of any threat
- Authorize the cheque book requests

User Portal:

It deals with the user activities. The user should be able to:

- Register or log in to the application to maintain a record of activities
- Deposit and withdraw money from the account
- View transactions and balance in the primary and savings account
- Transfer funds between different accounts and add recipients
- Request cheque books for different accounts

Pushing the code to GitHub Repository:

- Open Git Bash and navigate to the folder where you have created your files.
cd ICIN-Bank \ICIN-Bank-master
- Initialize the repository using the below command **git init**
- Add all the files to your git repository using the below command **git add .**
- Commit the changes using the below command **git commit -m "Initial commit"**
- Add the URL for the remote repository where your local repository will be pushed **git remote add origin https://github.com/Mahendra1272/ICIN-Bank \ICIN-Bank-master .git**
- Push the files to the folder you initially created using below command **git push -u origin master**

Links to the GitHub repository:

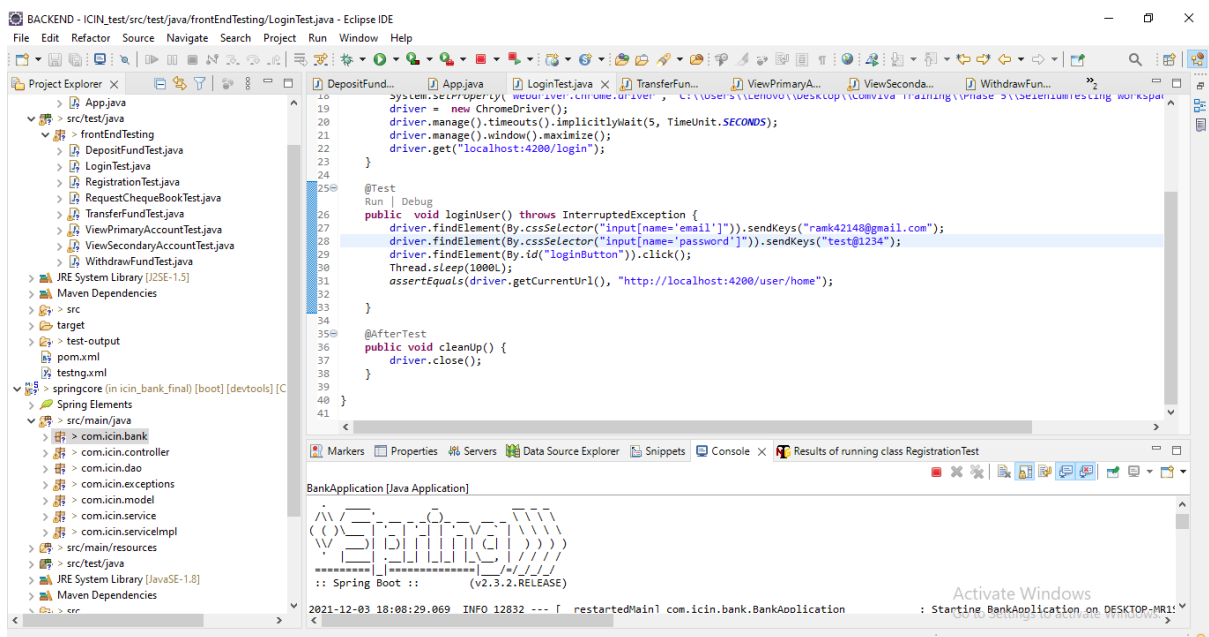
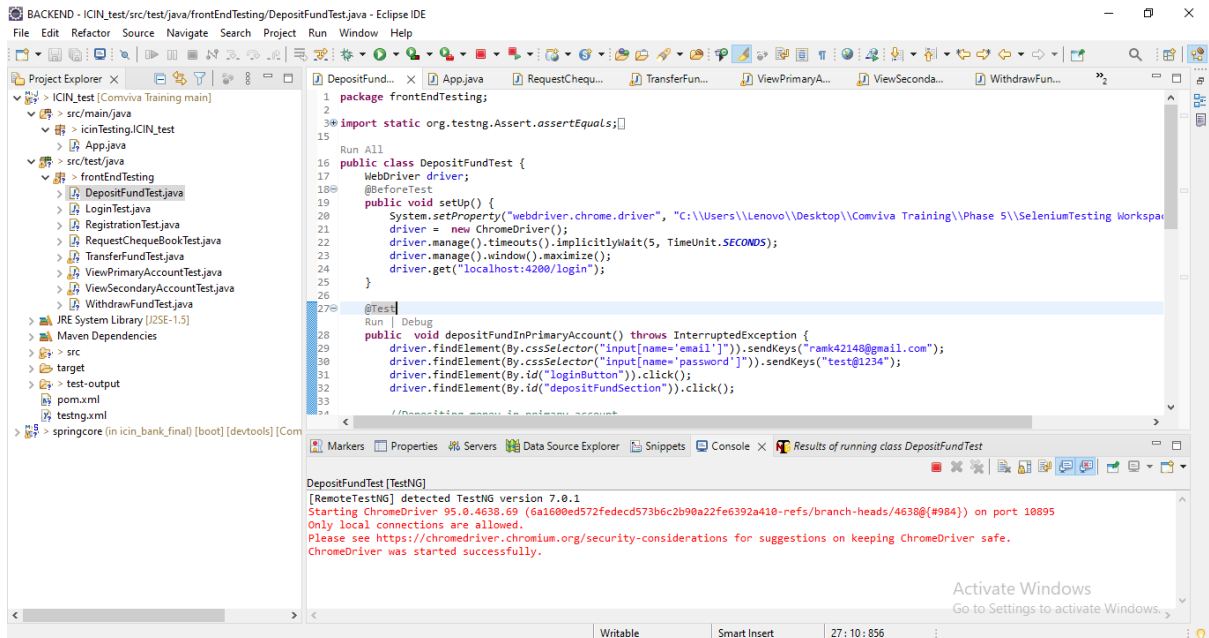
<https://github.com/Mahendra1272/ICIN-Bank.git>

Developer Details:

Mahendra Kumar Singh

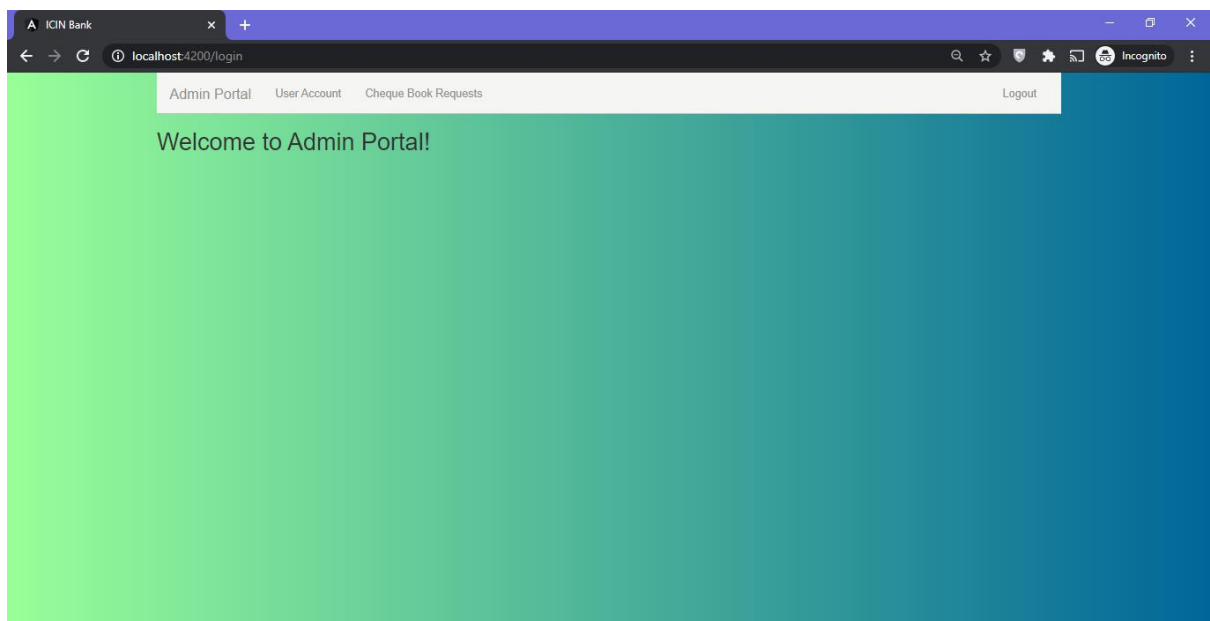
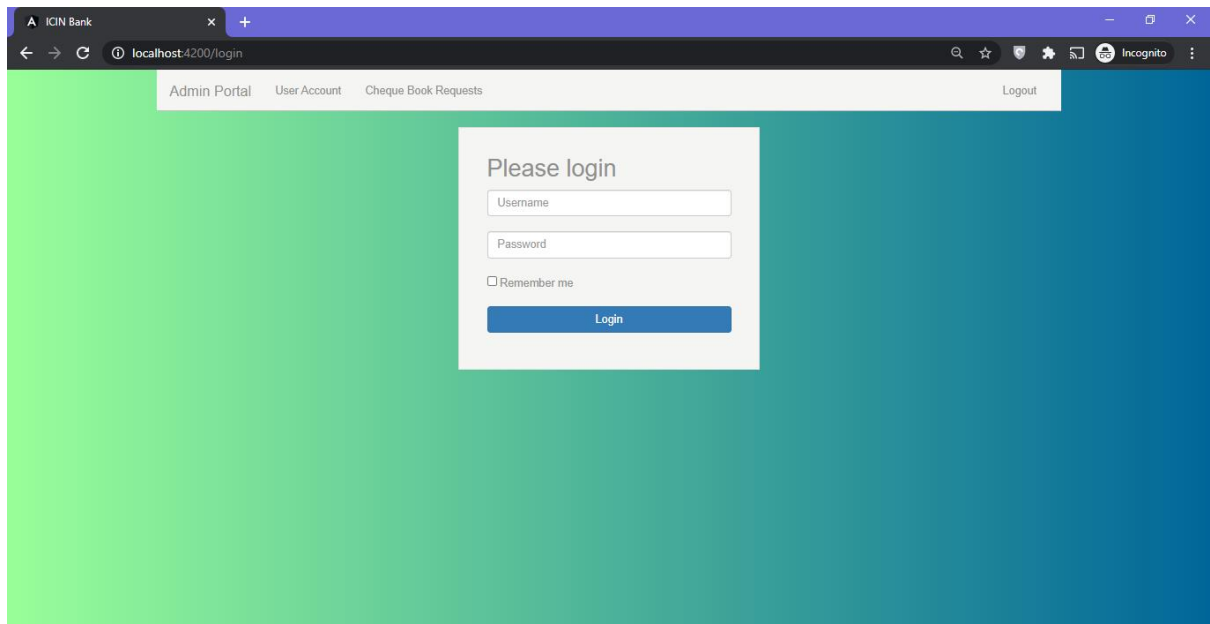
mahendrakumarsingh9893@gmail.com

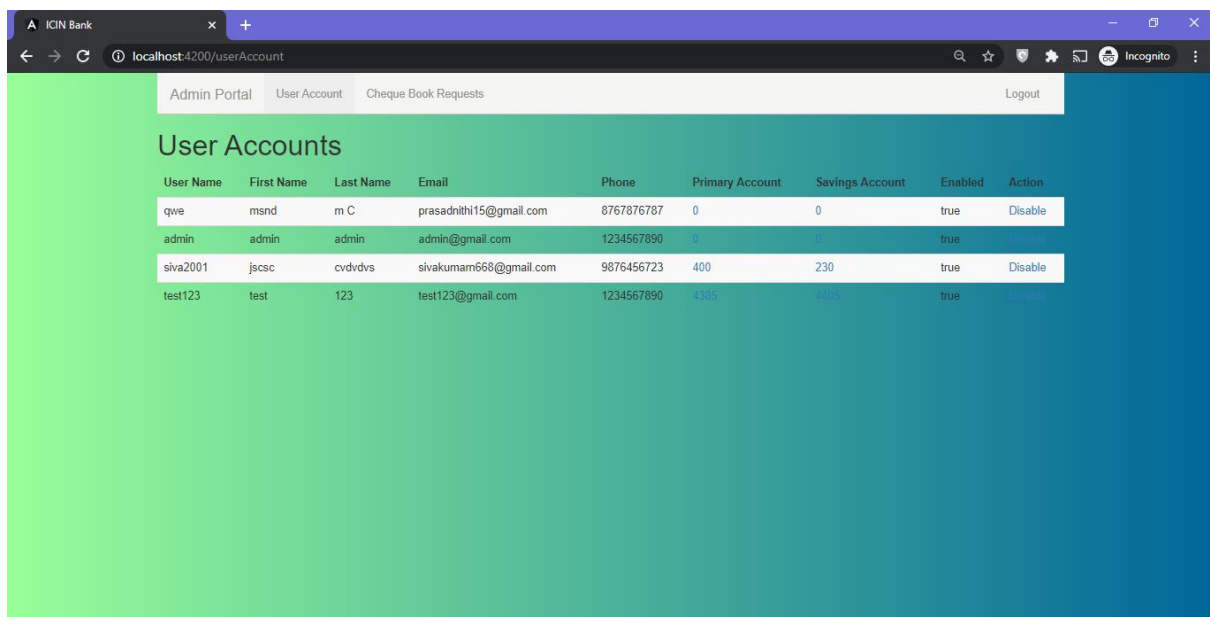
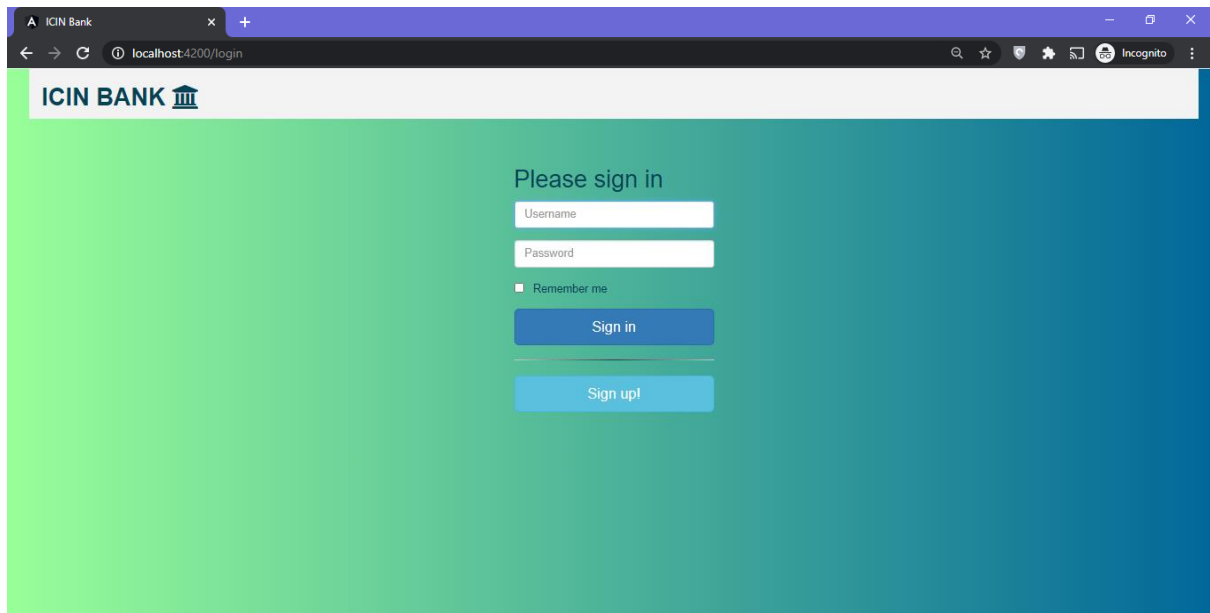
Screenshots and Source Code of all Scenarios in the Application:



```
BACKEND - ICIN_test/src/test/java/frontEndTesting/LoginTest.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Results of running class RegistrationTest
BankApplication [Java Application]
Hibernate:
    drop table if exists cheque_book
Hibernate:
    drop table if exists personal_transaction
Hibernate:
    drop table if exists primary_account
Hibernate:
    drop table if exists savings_account
Hibernate:
    drop table if exists transactions
Hibernate:
    drop table if exists user
Hibernate:
    create table admin (
      id integer not null auto_increment,
      emailid varchar(255),
      password varchar(255),
      primary key (id)
    ) engine=MyISAM
Hibernate:
    create table cheque_book (
      id bigint not null auto_increment,
      accepted bit not null,
      user_id bigint,
      primary key (id)
    ) engine=MyISAM
Hibernate:
    create table personal_transaction (
      transaction_id integer not null auto_increment,
```

```
BACKEND - ICIN_test/src/test/java/frontEndTesting/LoginTest.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Results of running class RegistrationTest
BankApplication [Java Application]
    ) engine=MyISAM
Hibernate:
    create table user (
      id bigint not null auto_increment,
      address varchar(255),
      fullname varchar(255),
      mailid varchar(255),
      password varchar(255),
      phonenumber bigint,
      surname varchar(255),
      primary_account_id bigint,
      savings_account_id bigint,
      primary key (id)
    ) engine=MyISAM
Hibernate:
    alter table cheque_book
      add constraint FKjhlwtt3o2tds933ybeah0web
      foreign key (user_id)
      references user (id)
Hibernate:
    alter table user
      add constraint FKbj8uoj9i40dory8wt45o3yb9n
      foreign key (primary_account_id)
      references primary_account (id)
Hibernate:
    alter table user
      add constraint FKihums7d3g5cv9ehminfs1539e
      foreign key (savings_account_id)
      references savings_account (id)
2021-12-03 18:08:50.158 INFO 12832 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.pl
2021-12-03 18:08:50.187 INFO 12832 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2021-12-03 18:08:50.221 INFO 12832 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2021-12-03 18:08:51.668 WARN 12832 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be per
2021-12-03 18:08:51.965 INFO 12832 --- [ restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
```





ICIN Bank

localhost:4200/login

HomeAccountsTransferRequestMe

Beneficiary Information

Name

Email

Phone

Account Number

Description

Add/Edit Recipient

List of Beneficiaries

Beneficiary Name	Beneficiary Email	Beneficiary Phone	Beneficiary Account Number	Description	
Test Recipient	testrecipient@gmail.com	9897654321	11223148	test description	delete

ICIN Bank

localhost:4200/chequeBookRequest

Admin PortalUser AccountCheque Book RequestsLogout

Cheque Book Requests

Request Id	User Name	Description	Confirmed?	Action
36	test123	kkakdda	false	Confirm
37	test123	er	false	

ICIN Bank

localhost:4200/login

Incognito

HomeAccountsTransferRequest

Me

My Profile

First Name

test

Last Name

123

Phone

1234567890

Your Email

test123@gmail.com

Username

test123

Change Settings

Your Account Information

Primary Account Number	Savings Account Number
11223152	11223153

ICIN Bank

localhost:4200/login

Incognito

HomeAccountsTransferRequest

Me

Request Cheque Book

Request Date and Time:

Description:

Description

Submit Request

October 2016

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Today

ICIN Bank

localhost:4200/login

Incognito

ICIN BANK

HomeAccountsTransferRequestMe

1. Please select the account you would like to transfer From:

-- select your account --

2. Please select the account you would like to transfer To:

-- select your account --

3. Please specify the amount you would like to transfer:

Amount Rs

Transfer

AssessmentGitHubICIN_BANK_CAPSTONE [Jenkins]

localhost:8080/job/ICIN_BANK_CAPSTONE/

log out

DashboardICIN_BANK_CAPSTONE

Back to Dashboard

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

GitHub

Rename

Pipeline Syntax

Build History

trend

find

Pipeline ICIN_BANK_CAPSTONE

Pipeline for the ICIN BANK Capstone Project

edit description

Disable Project

Recent Changes

Stage View

Average stage times:
(Average full run time: ~8min 41s)

	Declarative: Tool Install	Build	Install	Compile
#1 Dec 03 19:06 No Changes	2s	5min 1s	2min 18s	57s

Permalinks

Activate Windows
Go to Settings to activate Windows.

Practice Labs

AWS Management Console

+

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#

Services

Search for services, features, blogs, docs, and more

[Alt+S]

N. Virginia

Corestack_Role/m.madhugunike_comviva @ i-am-login

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Amazon S3

Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

Buckets (5)

Refresh

Copy ARN

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name	AWS Region	Access
<input type="radio"/>	angular-user	US East (N. Virginia) us-east-1	Objects can be public
<input type="radio"/>	elasticbeanstalk-us-east-2-483011108902	US East (Ohio) us-east-2	
<input type="radio"/>	icin-angular-admin	US East (N. Virginia) us-east-1	Objects can be public
<input type="radio"/>	icin-backend-admin	US East (N. Virginia) us-east-1	Objects can be public
<input type="radio"/>	icin-bank-backend-user	US East (N. Virginia) us-east-1	Objects can be public

Practice Labs

AWS Management Console

+

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#

Services

Search for services, features, blogs, docs, and more

[Alt+S]

N. Virginia

Corestack_Role/m.madhugunike_comviva @ i-am-login

Instances (3)

Filter instances

Instance state: running

Clear filters

Connect

Instance state

Actions

Launch instances

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	-	i-00719a24d460b10e3	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-236-19-173.co...	54.236.19.173
<input type="checkbox"/>	-	i-027ed5c8976c9318a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-221-16-83.com...	54.221.16.83
<input type="checkbox"/>	-	i-0672e48acadb1d18	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-227-191-46.co...	54.227.191.46

Select an instance above

Practice Labs x AWS Management Console x +

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia Corestack_Role/m.madhugunike_comviva @ i-am-login

Instances (3) Info

Filter instances

Instance state: running X Clear filters

Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
ec2-54-236-19-173.co...	54.236.19.173	—	—	disabled	launch-wizard-1	icinbank
ec2-54-221-16-83.com...	54.221.16.83	—	—	disabled	launch-wizard-2	icinbank
ec2-54-227-191-46.co...	54.227.191.46	—	—	disabled	launch-wizard-3	icinbank2

Select an instance above

Practice Labs x AWS Management Console x +

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia Corestack_Role/m.madhugunike_comviva @ i-am-login

RDS > Databases

Databases Group resources Modify Actions Restore from S3 Create database

Filter databases

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity
database-1	Instance	MySQL Community	us-east-1a	db.t2.micro	Available	2.95%	0 Connections

```
package icinTesting.ICIN_test;
```

```
/**
```

```
 * Hello world!
```

```
 */
```

```
*/
```

```
public class App
```

```
{
```

```
    public static void main( String[] args )
```

```
    {
```

```
        System.out.println( "Run the src/test/java" );
```

```
    }
```

```
}
```

```
package frontEndTesting;
```

```
import static org.testng.Assert.assertEquals;
```

```
import java.util.concurrent.TimeUnit;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.testng.annotations.AfterTest;
```

```
import org.testng.annotations.BeforeTest;
```

```
import org.testng.annotations.Test;
```

```
public class LoginTest {
```

```
    WebDriver driver;
```

@BeforeTest

```
public void setUp() {  
    System.setProperty("webdriver.chrome.driver",  
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting  
Workspace\\chromedriver.exe");  
    driver = new ChromeDriver();  
    driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);  
    driver.manage().window().maximize();  
    driver.get("localhost:4200/login");  
}
```

@Test

```
public void loginUser() throws InterruptedException {  
  
    driver.findElement(By.cssSelector("input[name='email']")).sendKeys("ramk42148@gmail.co  
m");  
  
    driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");  
    driver.findElement(By.id("loginButton")).click();  
    Thread.sleep(1000L);  
    assertEquals(driver.getCurrentUrl(), "http://localhost:4200/user/home");  
  
}
```

@AfterTest

```
public void cleanUp() {  
    driver.close();  
}  
  
}
```

package frontEndTesting;

```
import static org.testng.Assert.assertEquals;
```

```
import java.util.concurrent.TimeUnit;
```

```
import org.openqa.selenium.Alert;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.Select;
```

```
import org.testng.annotations.AfterTest;
```

```
import org.testng.annotations.BeforeTest;
```

```
import org.testng.annotations.Test;
```

```
public class DepositFundTest {
```

```
    WebDriver driver;
```

```
    @BeforeTest
```

```
    public void setUp() {
```

```
        System.setProperty("webdriver.chrome.driver",  
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting  
Workspace\\chromedriver.exe");
```

```
        driver = new ChromeDriver();
```

```
        driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
```

```
        driver.manage().window().maximize();
```

```
        driver.get("localhost:4200/login");
```

```
    }
```

```
    @Test
```

```
    public void depositFundInPrimaryAccount() throws InterruptedException {
```

```
        driver.findElement(By.cssSelector("input[name='email']")).sendKeys("ramk42148@gmail.co  
m");
```

```
        driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");
```

```

        driver.findElement(By.id("loginButton")).click();

        driver.findElement(By.id("depositFundSection")).click();

        //Depositing money in primary account
        driver.findElement(By.id("amountToBeDeposited")).sendKeys("1000");

        driver.findElement(By.id("depositMoneyButton")).click();
        Thread.sleep(1000L);
        Alert alert = driver.switchTo().alert();
        String message = alert.getText();
        assertEquals(message, "Money deposited successfully!!");
        alert.accept();
    }

    @Test
    public void depositFundInSavingAccount() throws InterruptedException {

        driver.findElement(By.id("depositFundSection")).click();
        Select accounts = new Select(driver.findElement(By.id("accountType")));
        accounts.selectByIndex(1);

        //Depositing money in saving account
        driver.findElement(By.id("amountToBeDeposited")).sendKeys("1000");

        driver.findElement(By.id("depositMoneyButton")).click();
        Thread.sleep(1000L);
        Alert alert = driver.switchTo().alert();
        String message = alert.getText();
        assertEquals(message, "Money deposited successfully!!");
        alert.accept();
    }

    @AfterTest

```



```

        public void cleanUp() {
            driver.close();
        }
    }

package frontEndTesting;

import static org.testng.Assert.assertEquals;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

public class RegistrationTest {
    WebDriver driver;

    @BeforeTest

    public void setUp() {
        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting
Workspace\\chromedriver.exe");

        driver = new ChromeDriver();

        driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

        driver.manage().window().maximize();

        driver.get("http://localhost:4200/register");
    }
}

```

@Test

public void registerUser() throws InterruptedException {

 //Personal Details

 driver.findElement(By.cssSelector("input[name='firstName']")).sendKeys("Test");

 driver.findElement(By.cssSelector("input[name='lastName']")).sendKeys("Test");

 driver.findElement(By.id("next")).click();

 assertEquals(driver.getCurrentUrl(), "http://localhost:4200/register/contactDetails");

 //Contact Details

 driver.findElement(By.cssSelector("input[name='email']")).sendKeys("test123@gmail.com");

 driver.findElement(By.cssSelector("input[name='phone']")).sendKeys("7788778877");

 driver.findElement(By.id("next")).click();

 assertEquals(driver.getCurrentUrl(), "http://localhost:4200/register/kycDetails");

 //Kyc Details

 driver.findElement(By.cssSelector("input[name='dob']")).click();

 driver.findElement(By.cssSelector("input[name='dob']")).sendKeys("02/02/2000");

 driver.findElement(By.cssSelector("input[name='pancard']")).sendKeys("GJOH19165");

 driver.findElement(By.id("next")).click();

 assertEquals(driver.getCurrentUrl(), "http://localhost:4200/register/addressDetails");

 //Address Details

 driver.findElement(By.cssSelector("input[name='addressLine1']")).sendKeys("C-1
Silver Oaks");

 driver.findElement(By.cssSelector("input[name='addressLine2']")).sendKeys("SV
Road");

 driver.findElement(By.cssSelector("input[name='addressLine3']")).sendKeys("Borivali");

 driver.findElement(By.cssSelector("input[name='addressLine4']")).sendKeys("Mumbai,
Maharashtra");

```

        driver.findElement(By.cssSelector("input[name='addressLine5']")).sendKeys("400067");

        driver.findElement(By.id("next")).click();

        assertEquals(driver.getCurrentUrl(),
"http://localhost:4200/register/userRegistration");

//User Registration

driver.findElement(By.cssSelector("input[name='userName']")).sendKeys("ramkrishna");

driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");

driver.findElement(By.cssSelector("input[name='confirm_password']")).sendKeys("test@123
4");

        driver.findElement(By.cssSelector("input[name='tnc']")).click();
        driver.findElement(By.cssSelector("button[id='signUpButton']")).click();
        Thread.sleep(2000L);
        assertEquals(driver.getCurrentUrl(), "http://localhost:4200/login");

    }

@Test
public void registerUser2() throws InterruptedException {

        driver.get("http://localhost:4200/register");

        driver.findElement(By.cssSelector("input[name='firstName']")).sendKeys("Rajesh");
        driver.findElement(By.cssSelector("input[name='lastName']")).sendKeys("kumar");
        driver.findElement(By.id("next")).click();
        assertEquals(driver.getCurrentUrl(), "http://localhost:4200/register/contactDetails");

//Contact Details

```

```
driver.findElement(By.cssSelector("input[name='email']").sendKeys("rajeshkumar@gmail.com");
```

```
driver.findElement(By.cssSelector("input[name='phone']").sendKeys("8877887788");
```

```
driver.findElement(By.id("next")).click();
```

```
assertEquals(driver.getCurrentUrl(), "http://localhost:4200/register/kycDetails");
```

```
//Kyc Details
```

```
driver.findElement(By.cssSelector("input[name='dob']").click();
```

```
driver.findElement(By.cssSelector("input[name='dob']").sendKeys("02/02/2000");
```

```
driver.findElement(By.cssSelector("input[name='pancard']").sendKeys("GJOH19165");
```

```
driver.findElement(By.id("next")).click();
```

```
assertEquals(driver.getCurrentUrl(), "http://localhost:4200/register/addressDetails");
```

```
//Address Details
```

```
driver.findElement(By.cssSelector("input[name='addressLine1']").sendKeys("C-1 Silver Oaks");
```

```
driver.findElement(By.cssSelector("input[name='addressLine2']").sendKeys("SV Road");
```

```
driver.findElement(By.cssSelector("input[name='addressLine3']").sendKeys("Borivali");
```

```
driver.findElement(By.cssSelector("input[name='addressLine4']").sendKeys("Mumbai, Maharashtra");
```

```
driver.findElement(By.cssSelector("input[name='addressLine5']").sendKeys("400067");
```

```
driver.findElement(By.id("next")).click();
```

```
assertEquals(driver.getCurrentUrl(),  
"http://localhost:4200/register/userRegistration");
```

```
//User Registration
```

```
driver.findElement(By.cssSelector("input[name='userName']").sendKeys("ramkrishna");
```

```
driver.findElement(By.cssSelector("input[name='password']").sendKeys("test@1234");
```

```
4");
    driver.findElement(By.cssSelector("input[name='confirm_password']").sendKeys("test@123
```

```
    driver.findElement(By.cssSelector("input[name='tnc']").click();
```

```
    driver.findElement(By.cssSelector("button[id='signUpButton']").click();
```

```
    Thread.sleep(2000L);
```

```
    assertEquals(driver.getCurrentUrl(), "http://localhost:4200/login");
```

```
    }
```

```
    @AfterTest
```

```
    public void cleanUp() {
```

```
        driver.close();
```

```
    }
```

```
}
```

```
package frontEndTesting;
```

```
import static org.testng.Assert.assertEquals;
```

```
import java.util.concurrent.TimeUnit;
```

```
import org.openqa.selenium.Alert;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.testng.annotations.AfterTest;
```

```
import org.testng.annotations.BeforeTest;
```

```
import org.testng.annotations.Test;
```

```
public class RequestChequeBookTest {
```

```

WebDriver driver;

@BeforeTest

public void setUp() {

    System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting
Workspace\\chromedriver.exe");

    driver = new ChromeDriver();

    driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

    driver.manage().window().maximize();

    driver.get("localhost:4200/login");

}

@Test

public void requestChequeBook() throws InterruptedException {

driver.findElement(By.cssSelector("input[name='email']")).sendKeys("ramk42148@gmail.co
m");

driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");

    driver.findElement(By.id("loginButton")).click();

    driver.findElement(By.id("chequebookRequestLink")).click();

    driver.findElement(By.id("chequebookRequestPrimary")).click();

    driver.findElement(By.xpath("//button[text()='Submit Request']")).click();

    Thread.sleep(1000L);

    Alert alert = driver.switchTo().alert();

    String msg = alert.getText();

    alert.accept();

    assertEquals(msg, "Chequebook request submitted");

}

@AfterTest

public void cleanUp() {

```

```

        driver.close();
    }

}

package frontEndTesting;

import static org.testng.Assert.assertEquals;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import frontEndTesting.LoginTest;

public class TransferFundTest {

    WebDriver driver;

    @BeforeTest

    public void setUp() {

        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting
Workspace\\chromedriver.exe");

        driver = new ChromeDriver();

        driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

        driver.manage().window().maximize();

        driver.get("localhost:4200");
    }
}

```

```
}
```

```
@Test
```

```
public void transferFund() throws InterruptedException {
```

```
driver.findElement(By.cssSelector("input[name='email']")).sendKeys("ramk42148@gmail.co  
m");
```

```
driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");
```

```
driver.findElement(By.id("loginButton")).click();
```

```
driver.findElement(By.linkText("Transfer")).click();;
```

```
//Filling Up transfer fund form
```

```
driver.findElement(By.id("beneficiaryName")).sendKeys("Rajesh kumar");
```

```
driver.findElement(By.id("beneficiaryAccountNumber")).sendKeys("22113346");
```

```
driver.findElement(By.id("beneficiaryIFSCCode")).sendKeys("ICIN00105");
```

```
driver.findElement(By.id("amountToBeTransferred")).sendKeys("100");
```

```
driver.findElement(By.id("description")).sendKeys("Fund Transfer");
```

```
driver.findElement(By.id("transferFundButton")).click();
```

```
Thread.sleep(2000L);
```

```
Alert alert = driver.switchTo().alert();
```

```
String text = alert.getText();
```

```
System.out.println(text);
```

```
alert.accept();
```

```
assertEquals(text, "Transaction Successful");
```

```
driver.navigate().back();
```

```
}
```

```
@Test
```



```

public void transferFundGreaterThanBalance() throws InterruptedException {

    driver.findElement(By.linkText("Transfer")).click();

    //Filling Up transfer fund form
    driver.findElement(By.id("beneficiaryName")).sendKeys("Rajesh kumar");
    driver.findElement(By.id("beneficiaryAccountNumber")).sendKeys("22113346");
    driver.findElement(By.id("beneficiaryIFSCCode")).sendKeys("ICIN00105");
    driver.findElement(By.id("amountToBeTransferred")).sendKeys("10000");
    driver.findElement(By.id("description")).sendKeys("Fund Transfer");
    driver.findElement(By.id("transferFundButton")).click();
    Thread.sleep(2000L);
    Alert alert = driver.switchTo().alert();
    String text = alert.getText();
    System.out.println(text);
    alert.accept();
    assertEquals(text, "Insufficient funds");
    driver.navigate().back();

}

```

@Test

```

public void transferFundWithIncorrectAccountNumber() throws InterruptedException {

    driver.findElement(By.linkText("Transfer")).click();

    //Filling Up transfer fund form
    driver.findElement(By.id("beneficiaryName")).sendKeys("Rajesh kumar");
    driver.findElement(By.id("beneficiaryAccountNumber")).sendKeys("221133467");
    driver.findElement(By.id("beneficiaryIFSCCode")).sendKeys("ICIN00105");
    driver.findElement(By.id("amountToBeTransferred")).sendKeys("10");
}

```

```

        driver.findElement(By.id("description")).sendKeys("Fund Transfer");

        driver.findElement(By.id("transferFundButton")).click();

        Thread.sleep(2000L);

        Alert alert = driver.switchTo().alert();

        String text = alert.getText();

        System.out.println(text);

        alert.accept();

        //assertEquals(text, "Transaction failed Incorrect beneficiary account number");

        driver.navigate().back();

    }

```

```

        @AfterTest

        public void cleanUp() {

            driver.close();

        }

    }

```

```

package frontEndTesting;

```

```

import static org.testng.Assert.assertEquals;

```

```

import java.util.concurrent.TimeUnit;

```

```

import org.openqa.selenium.Alert;

```

```

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.WebDriver;

```

```

import org.openqa.selenium.chrome.ChromeDriver;

```

```

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;


public class ViewPrimaryAccountTest {

    WebDriver driver;

    @BeforeTest

    public void setUp() {

        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting
Workspace\\chromedriver.exe");

        driver = new ChromeDriver();

        driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

        driver.manage().window().maximize();

        driver.get("localhost:4200/login");

    }


    @Test

    public void requestChequeBook() throws InterruptedException {

        driver.findElement(By.cssSelector("input[name='email']")).sendKeys("ramk42148@gmail.co
m");

        driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");

        driver.findElement(By.id("loginButton")).click();

        driver.findElement(By.id("viewPrimaryAccountDetails")).click();

        driver.navigate().back();

        driver.findElement(By.id("viewPrimaryAccountDetails")).click();

        //viewSavingAccountDetails

    }

```

```

        @AfterTest

        public void cleanUp() {

            driver.close();

        }

    }

package frontEndTesting;

import static org.testng.Assert.assertEquals;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

public class ViewSecondaryAccountTest {

    WebDriver driver;

    @BeforeTest

    public void setUp() {

        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting
Workspace\\chromedriver.exe");

        driver = new ChromeDriver();

```

```

        driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

        driver.manage().window().maximize();

        driver.get("localhost:4200/login");

    }

    @Test

    public void requestChequeBook() throws InterruptedException {

        driver.findElement(By.cssSelector("input[name='email']")).sendKeys("ramk42148@gmail.co
m");

        driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");

        driver.findElement(By.id("loginButton")).click();

        driver.findElement(By.id("viewSavingAccountDetails")).click();

        driver.navigate().back();

    }

    @AfterTest

    public void cleanUp() {

        driver.close();

    }

}

```

```

package frontEndTesting;

```

```
import static org.testng.Assert.assertEquals;
```

```
import java.util.concurrent.TimeUnit;
```

```
import org.openqa.selenium.Alert;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.Select;
```

```
import org.testng.annotations.AfterTest;
```

```
import org.testng.annotations.BeforeTest;
```

```
import org.testng.annotations.Test;
```

```
public class WithdrawFundTest {
```

```
    WebDriver driver;
```

```
    @BeforeTest
```

```
    public void setUp() {
```

```
        System.setProperty("webdriver.chrome.driver",  
"C:\\Users\\Lenovo\\Desktop\\Comviva Training\\Phase 5\\SeleniumTesting  
Workspace\\chromedriver.exe");
```

```
        driver = new ChromeDriver();
```

```
        driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
```

```
        driver.manage().window().maximize();
```

```
        driver.get("localhost:4200/login");
```

```
    }
```

```
    @Test
```

```
    public void withdrawFundInPrimaryAccount() throws InterruptedException {
```

```
        driver.findElement(By.cssSelector("input[name='email']")).sendKeys("ramk42148@gmail.co  
m");
```

```

driver.findElement(By.cssSelector("input[name='password']")).sendKeys("test@1234");

driver.findElement(By.id("loginButton")).click();

driver.findElement(By.id("withdrawFudSection")).click();

//Depositing money in primary account
driver.findElement(By.id("amountToWithdrawn")).sendKeys("100");

driver.findElement(By.id("withdrawMoneyButton")).click();
Thread.sleep(1000L);
Alert alert = driver.switchTo().alert();
String message = alert.getText();
assertEquals(message, "Money Withdrawn successfully!!");
alert.accept();
}

@Test
public void withdrawFundInSavingAccount() throws InterruptedException {

driver.findElement(By.id("withdrawFudSection")).click();

Select accounts = new Select(driver.findElement(By.id("accountType")));
accounts.selectByIndex(1);

//Depositing money in saving account
driver.findElement(By.id("amountToWithdrawn")).sendKeys("100");

driver.findElement(By.id("withdrawMoneyButton")).click();
Thread.sleep(1000L);
Alert alert = driver.switchTo().alert();
String message = alert.getText();
assertEquals(message, "Money Withdrawn successfully!!");
alert.accept();
}

```

@Test

```
public void withdrawingAmountGreaterThanCurrentAmountForSavingsAccount() throws  
InterruptedException {
```

```
    Thread.sleep(1000L);  
    driver.navigate().back();  
    driver.findElement(By.id("withdrawFudSection")).click();  
    Select accounts = new Select(driver.findElement(By.id("accountType")));  
    accounts.selectByIndex(1);  
    //Depositing money in saving account  
    driver.findElement(By.id("amountToWithdrawn")).sendKeys("10000");  
  
    driver.findElement(By.id("withdrawMoneyButton")).click();  
    Thread.sleep(1000L);  
    Alert alert = driver.switchTo().alert();  
    String message = alert.getText();  
    assertEquals(message, "Insufficient Funds");  
    alert.accept();
```

```
}
```

@Test

```
public void withdrawingAmountGreaterThanCurrentAmountForPrimaryAccount() throws  
InterruptedException {
```

```
    driver.findElement(By.id("withdrawFudSection")).click();  
  
    //Depositing money in saving account  
    driver.findElement(By.id("amountToWithdrawn")).sendKeys("10000");  
  
    driver.findElement(By.id("withdrawMoneyButton")).click();  
    Thread.sleep(1000L);  
    Alert alert = driver.switchTo().alert();  
    String message = alert.getText();  
    assertEquals(message, "Insufficient Funds");
```



```
        alert.accept();
    }

    @AfterTest
    public void cleanUp() {
        driver.close();
    }
}
```