*Term Project*

*Instructor: Faisal Khan*

|  |  |  |
| --- | --- | --- |
| *Name* | *Student ID* | *Group#* |
| Poliana de Barros Pedro | C0766569 |  |
| Anjali Pandiri | C0765031 |  |
| Mounica Chidepudi | C0761734 |  |
| Tutku ramazanoglu | C0768385 |  |

1-Use Documentation Comments where needed and use Javadoc to create HTML files for documentation.

2-In your project src folder add this word document with all your group members names and paste the screen shots of your output after each question.

3-Right Click on your Project Folder in Eclipse, Copy Option is available. Create a copy of the Project and upload the zipped folder using Moodle.

Create a web based application using Either JSF or JSPs & Servlets . Your web application should maintain

Bank Account Records. You can use a model class BankAccount

with fields like.

Account ID, Account Type(Checking/Saving), AccouHolderName,

AccountHolderContact, AccountBalance,AccountSetupDate.

You should be able to perform all these CRUD Operations.

a)Create a Database BankAccounts for your application.

Create a table named Accounts in this database.

Insert 5 to 6 Records in this table. 10

Scripts:

Create database

CREATE DATABASE `BankAccount`;

CREATE TABLE `Accounts` (

`account\_id` int(11) NOT NULL,

`type` varchar(1) DEFAULT NULL,

`holder\_name` varchar(45) DEFAULT NULL,

`holder\_contact` varchar(45) DEFAULT NULL,

`balance` decimal(6,2) DEFAULT NULL,

`setup\_date` varchar(45) DEFAULT NULL,

PRIMARY KEY (`account\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `BankAccount`.`Accounts` (`account\_id`, `type`, `holder\_name`, `holder\_contact`, `balance`, `setup\_date`)

VALUES (101, 'S', 'Maria', '74384734', '5020.90', '2020-20-02');

INSERT INTO `BankAccount`.`Accounts` (`account\_id`, `type`, `holder\_name`, `holder\_contact`, `balance`, `setup\_date`)

VALUES (102, 'C', 'Peter', '345684734', '3000.80', '2020-13-08');

INSERT INTO `BankAccount`.`Accounts` (`account\_id`, `type`, `holder\_name`, `holder\_contact`, `balance`, `setup\_date`)

VALUES ('103', 'S', 'John', '14344734', '5000.50', '2020-10-12');

INSERT INTO `BankAccount`.`Accounts` (`account\_id`, `type`, `holder\_name`, `holder\_contact`, `balance`, `setup\_date`)

VALUES ('104', 'S', 'Bob', '24374799', '1000', '2020-30-06');

INSERT INTO `BankAccount`.`Accounts` (`account\_id`, `type`, `holder\_name`, `holder\_contact`, `balance`, `setup\_date`)

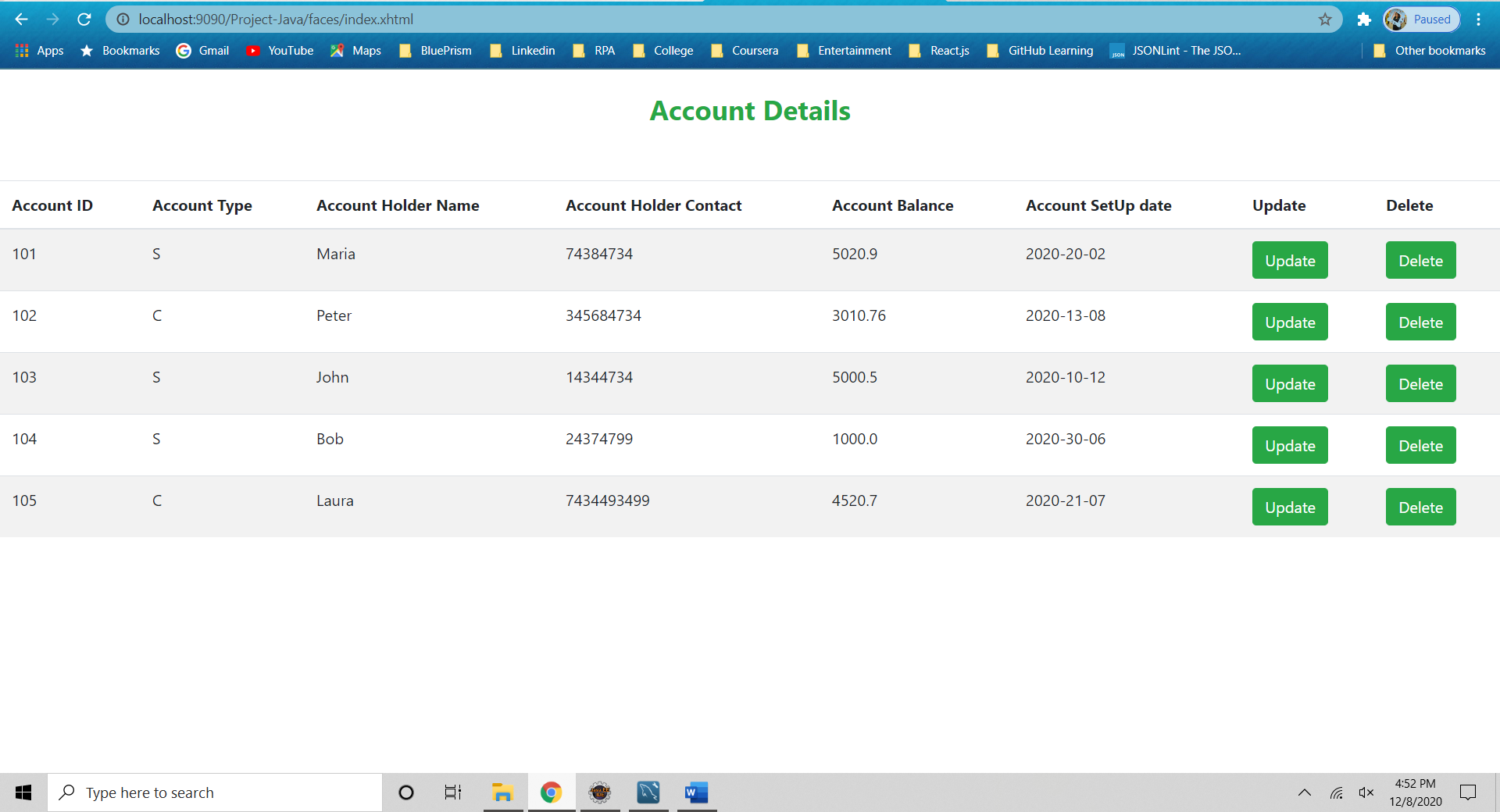
VALUES ('105', 'C', 'Laura', '74344934', '4520.70', '2020-21-07');

Output:

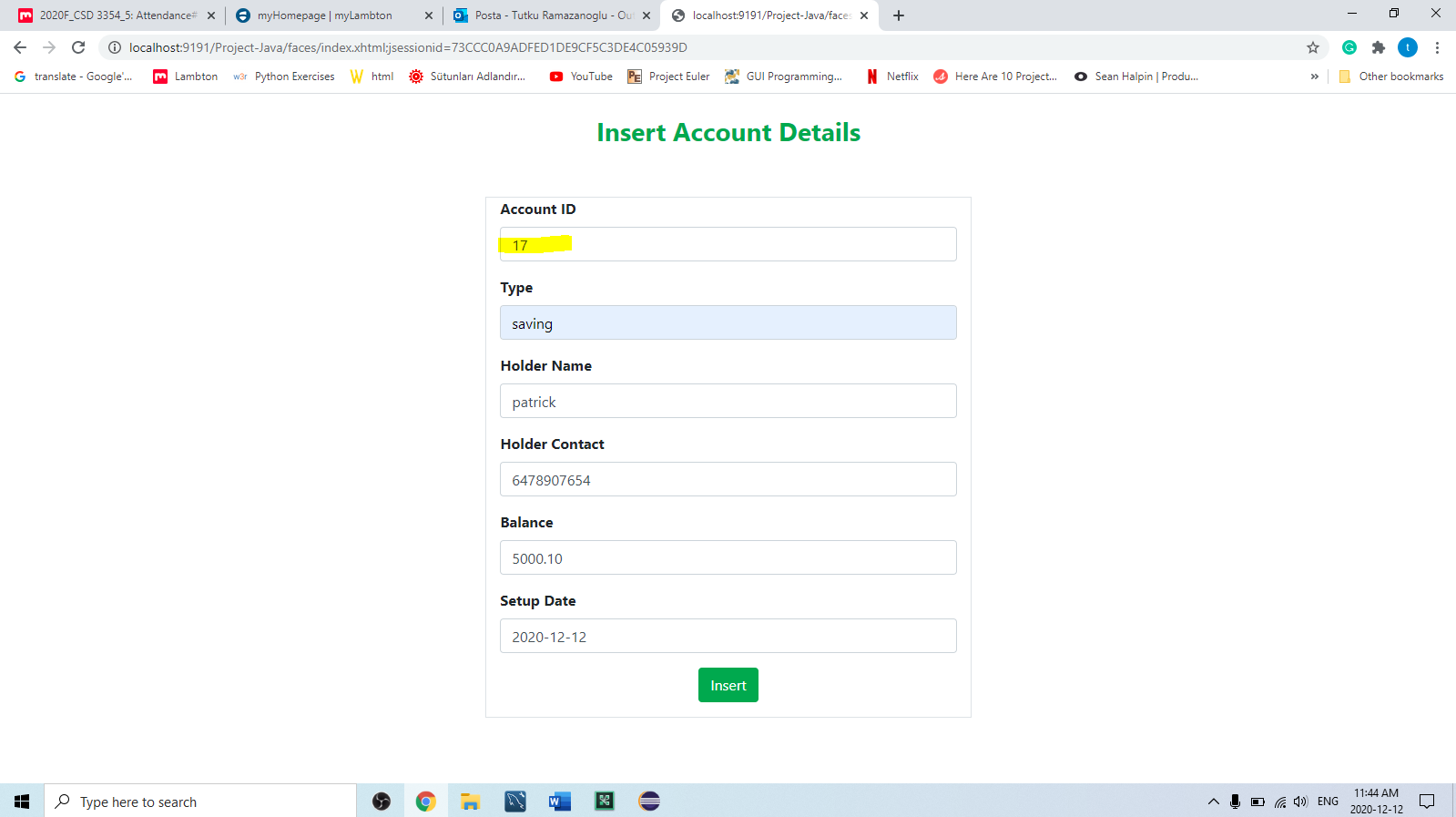
Graphical user interface, text, email

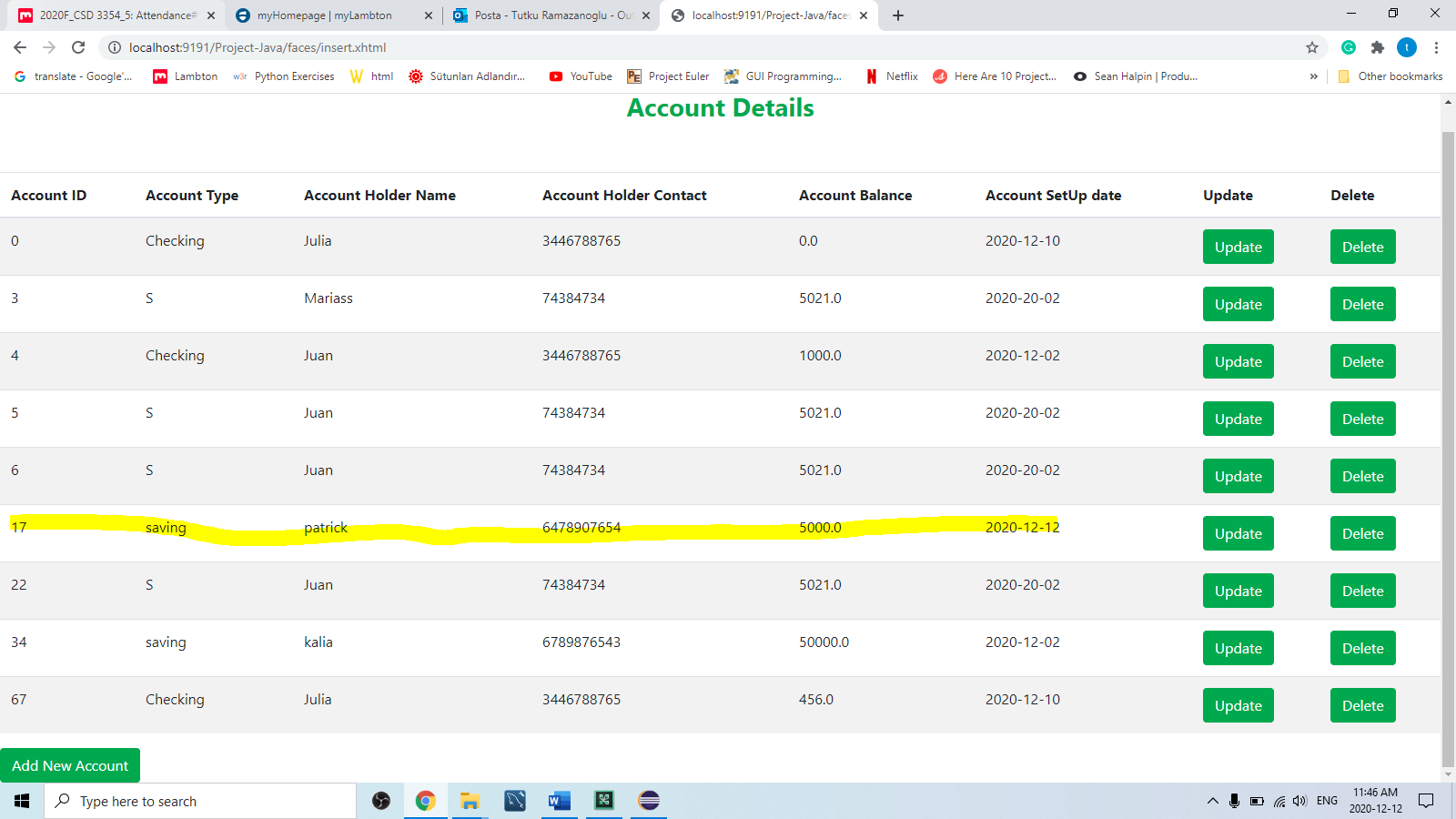
Description automatically generated

b) Display all the Bank Accounts in your xhtml page View or a JSP page as a table. 20

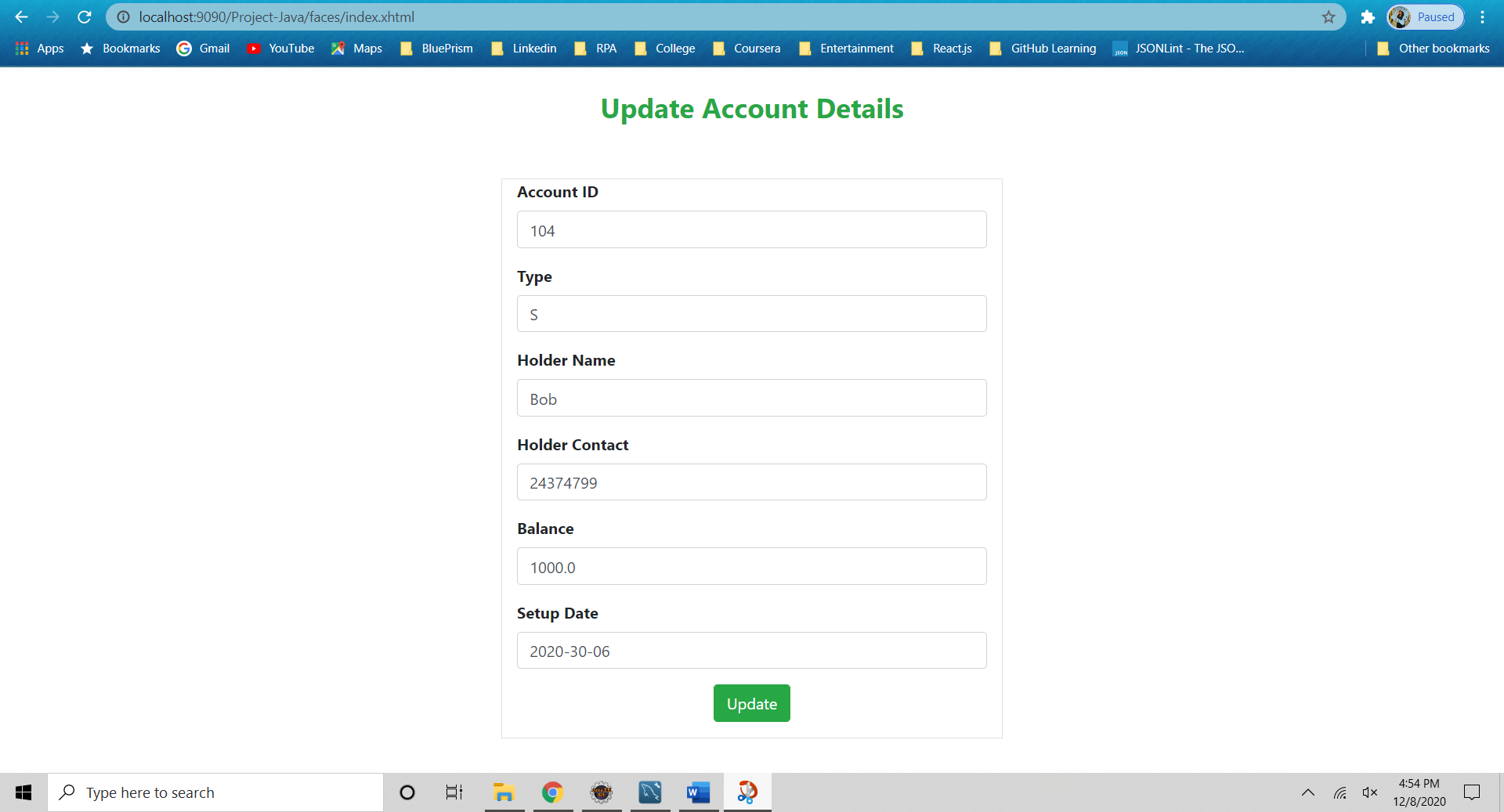


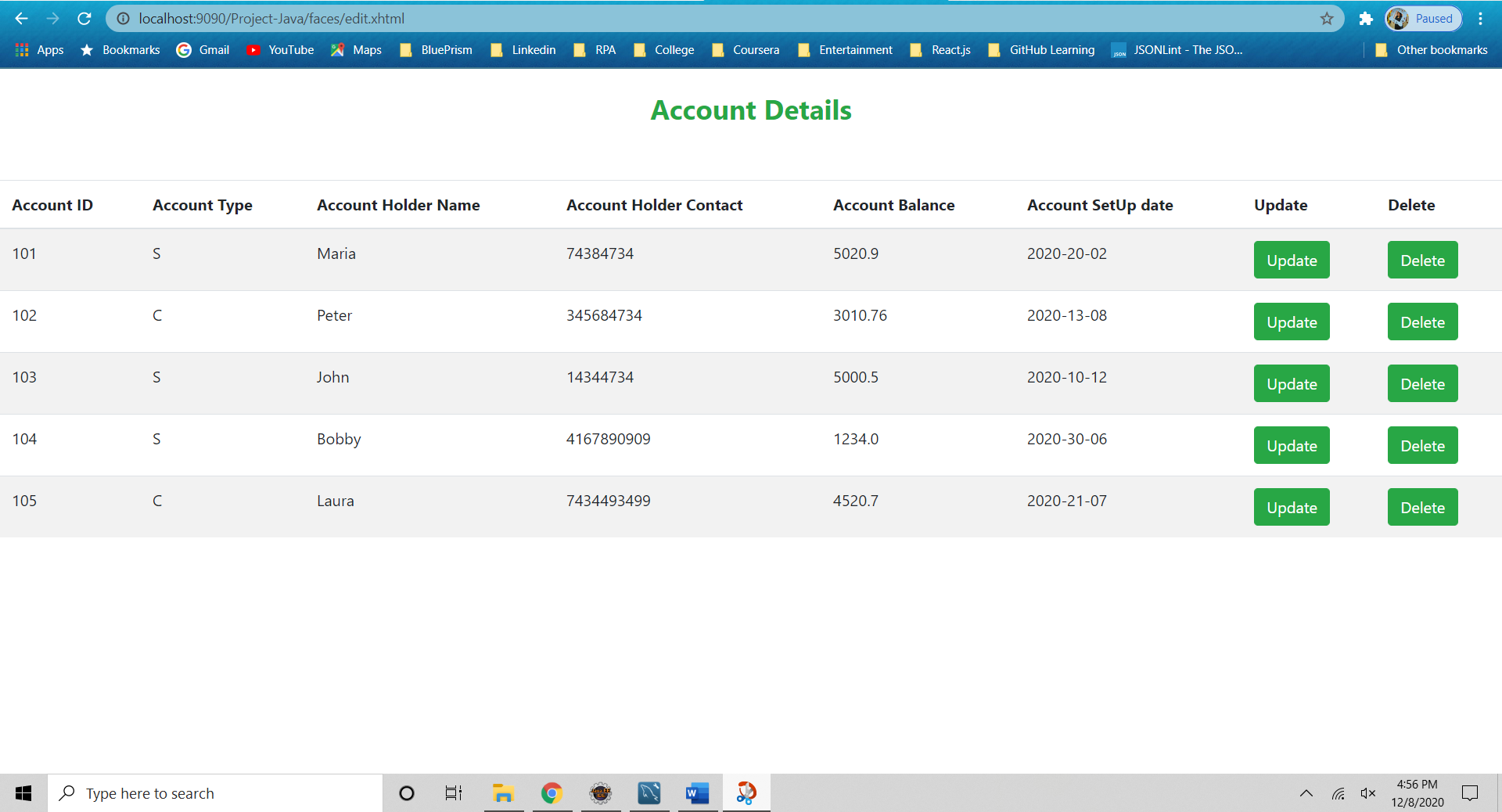
c) Add a new Bank Account in to the database using a button on your view page. 20



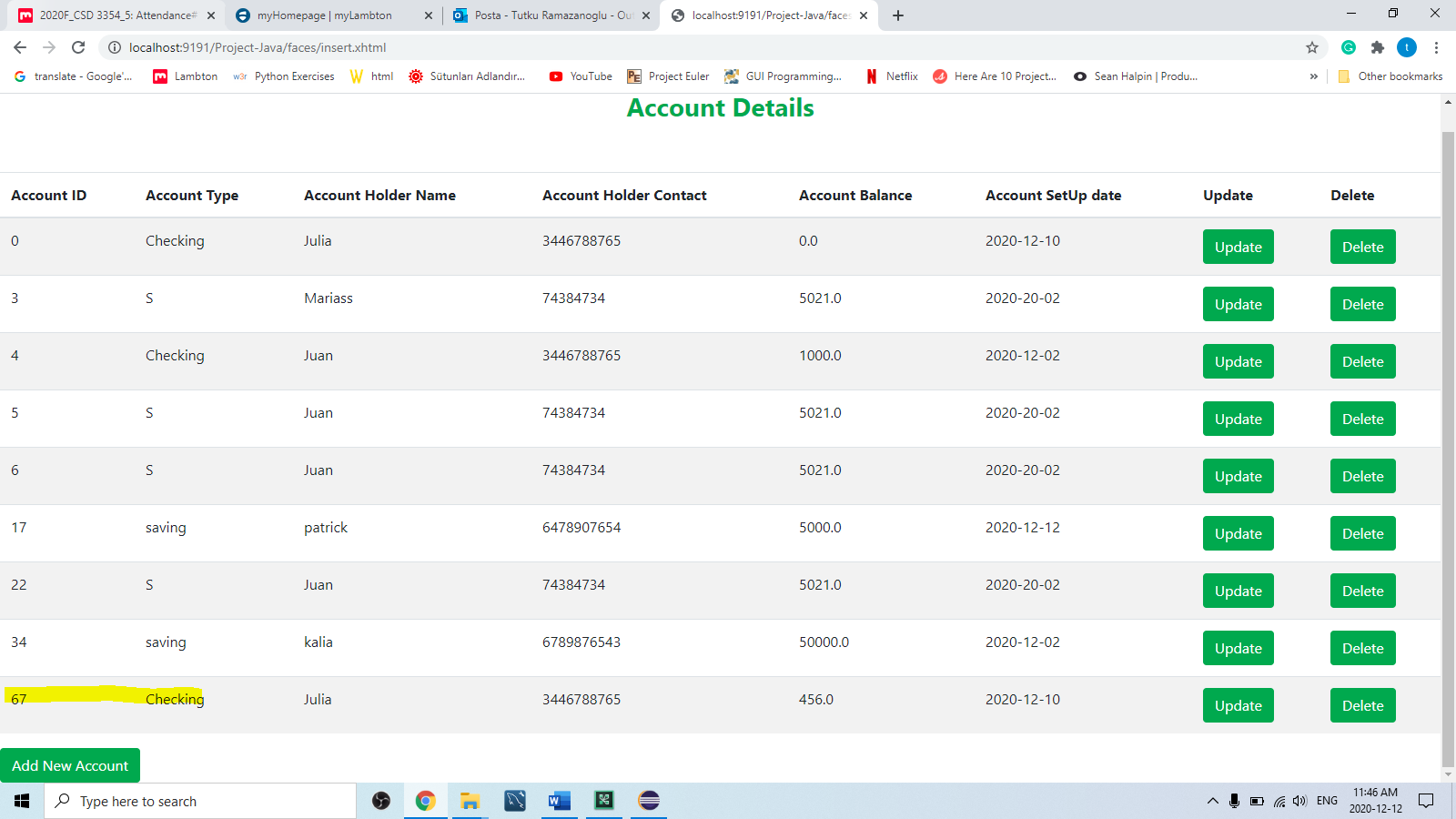


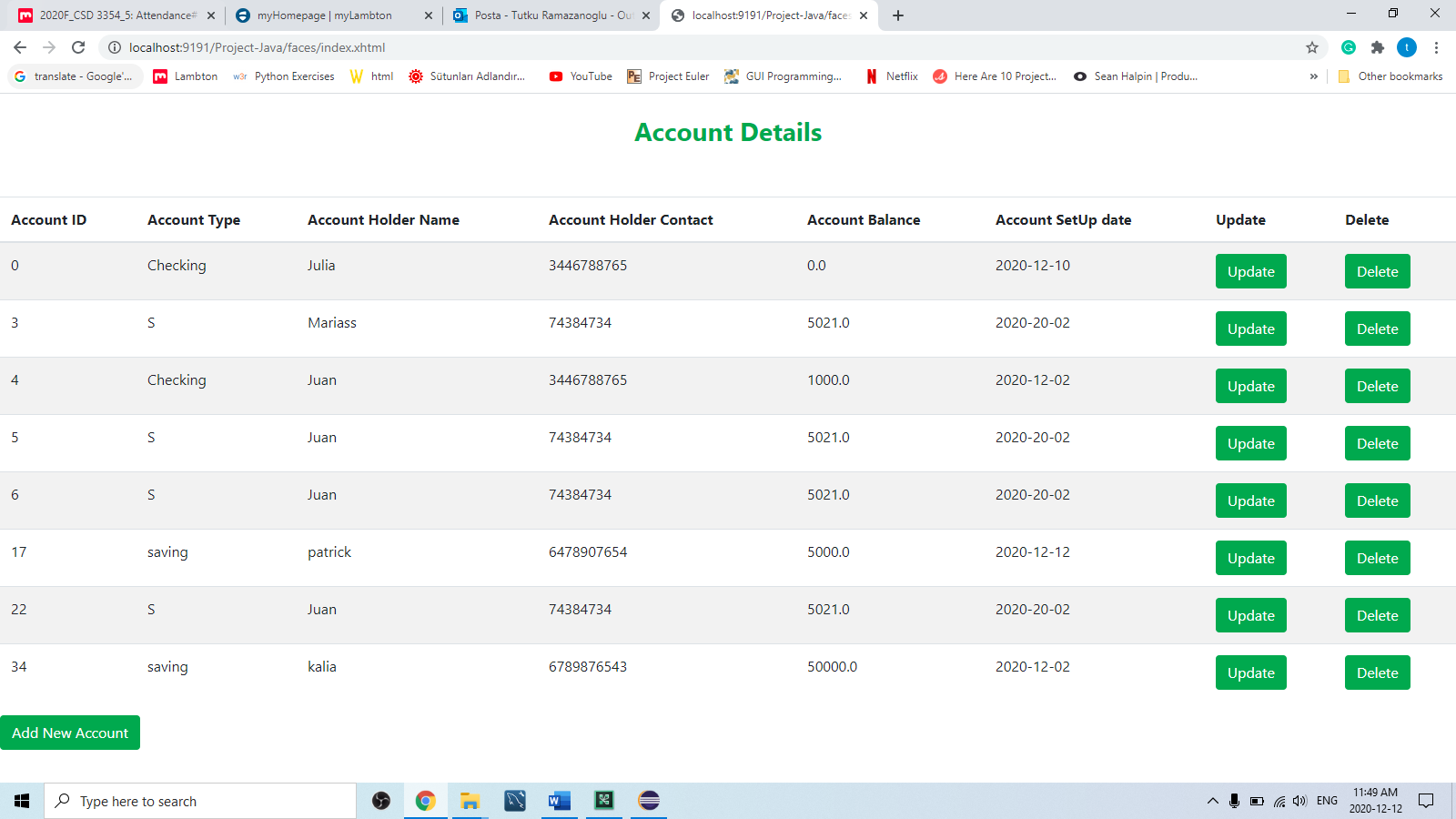
d)Update an existing Bank Account using an Update link on your view page. 20





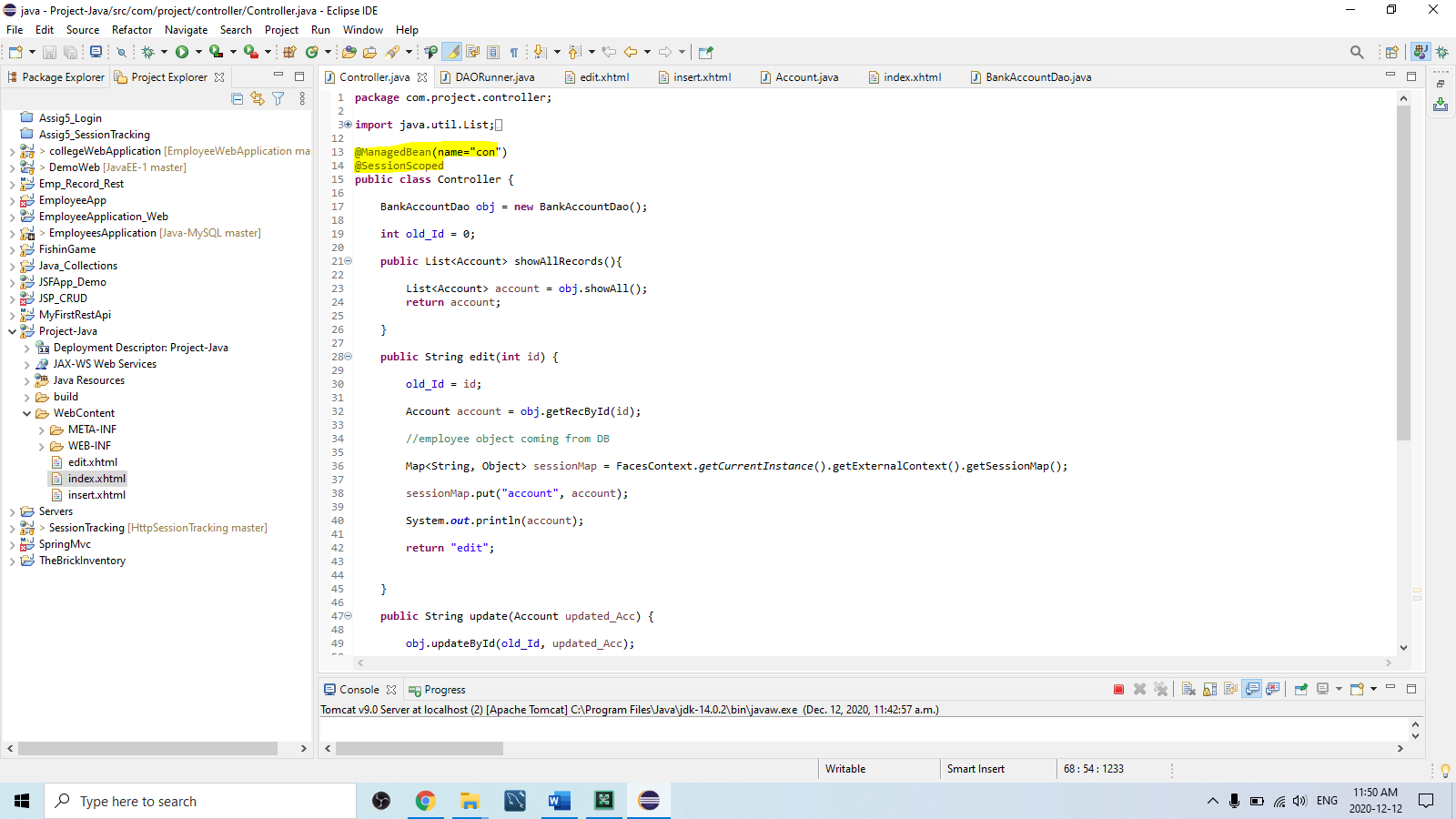
e)Delete a Bank Account using a Delete Link on your view page. 20

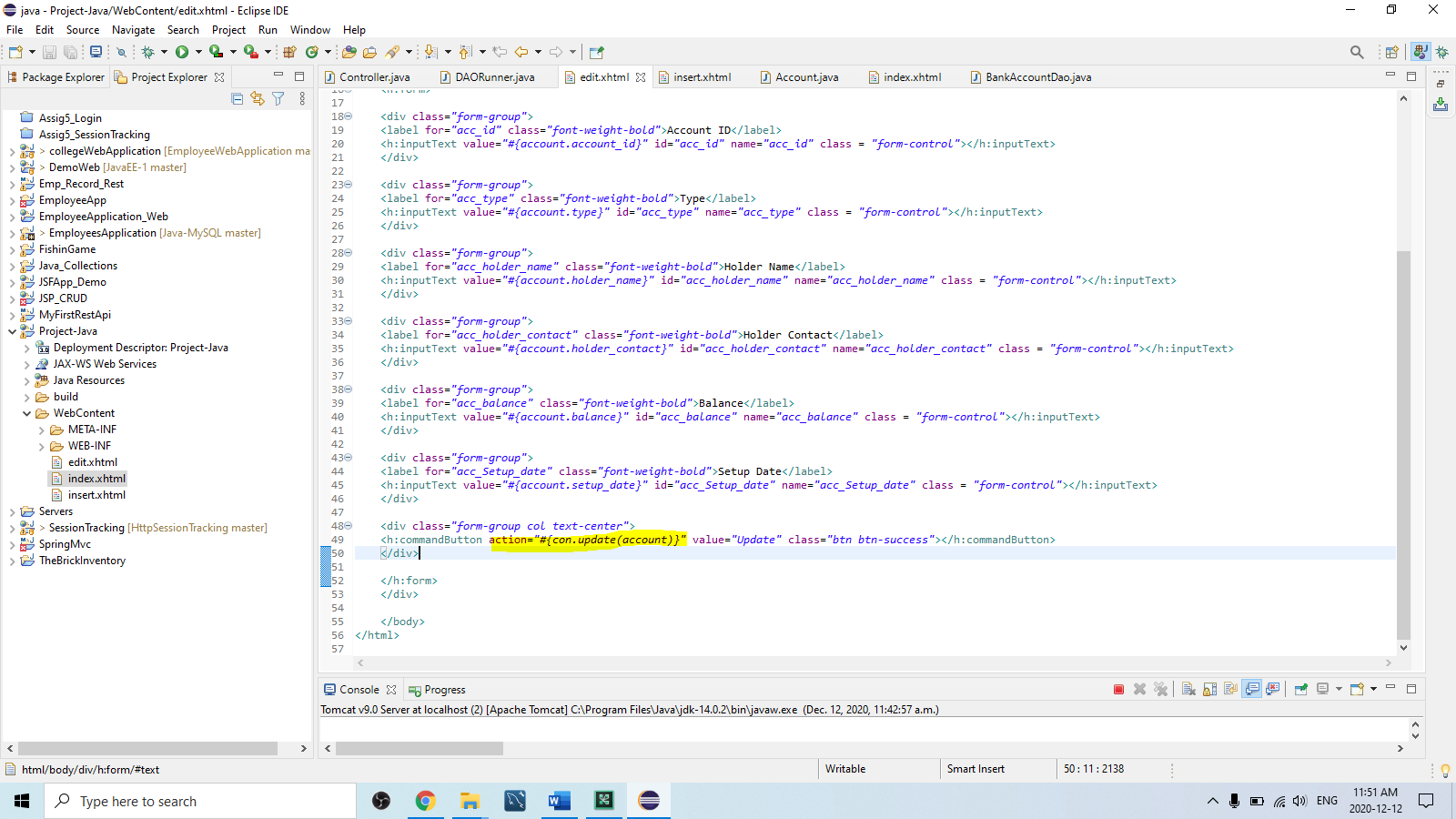




f) Use ManagedBean if using JSF with annotation and create separate DAO class for Crud operations or use a Simple Java Bean as your Model class if using JSPs and Servlets.Try to create separate methods for all crud operations. 10

-Edit the account using managedBeans





-Insert to new account using new\_acc managedBeans

