Even Semester (2019)



**BINUS UNIVERSITY**



**BINUS INTERNATIONAL**



**Assignment Cover Letter**

**(Individual Work)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  |  |  |
| **Student Information**: | **Surname** | | |  |  | **Given Names**  **Fauzan** | **Student ID Number**  **2201798326** |
| 1. | **Arief** | |
|  |  |  |  |  |  |  |  |
| **Course Code** | **: COMP6510** |  |  |  |  | **Course Name** | **: Programming Language** |
| **Class** | **: L2BC** |  |  |  |  | **Name of Lecturer(s)** | : **MINALDI LOEIS** |
|  |  |  |  |  |  |  |  |
| **Major** | **: CS** |  |  |  |  |  |  |
| **Title of Assignment**  (if any) | : **Vote Anything** |  |  |  |  |  |  |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** | |  |  |  |  |  |
| **Due Date** | **: 01 - 07 - 2019** | |  |  |  | **Submission Date** | **: 01 – 07 -2019** |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

**Fauzan Athallah Arief**

**II. Table of Contents**

1. **Cover**
2. **Table of Content**
3. **Project Specifications**
4. **Solution Design**
5. **Code Implementation Discussion**
6. **Implementation**
7. **How it works**
8. **Detailed code description**
9. **Evidence of Working Programme**
   * 1. **Login page**
     2. **Home Page**
     3. **Create Vote**
     4. **Vote Page**
     5. **Check Vote**
     6. **Result Page**
10. **Credit**

**III. Project Specifications**

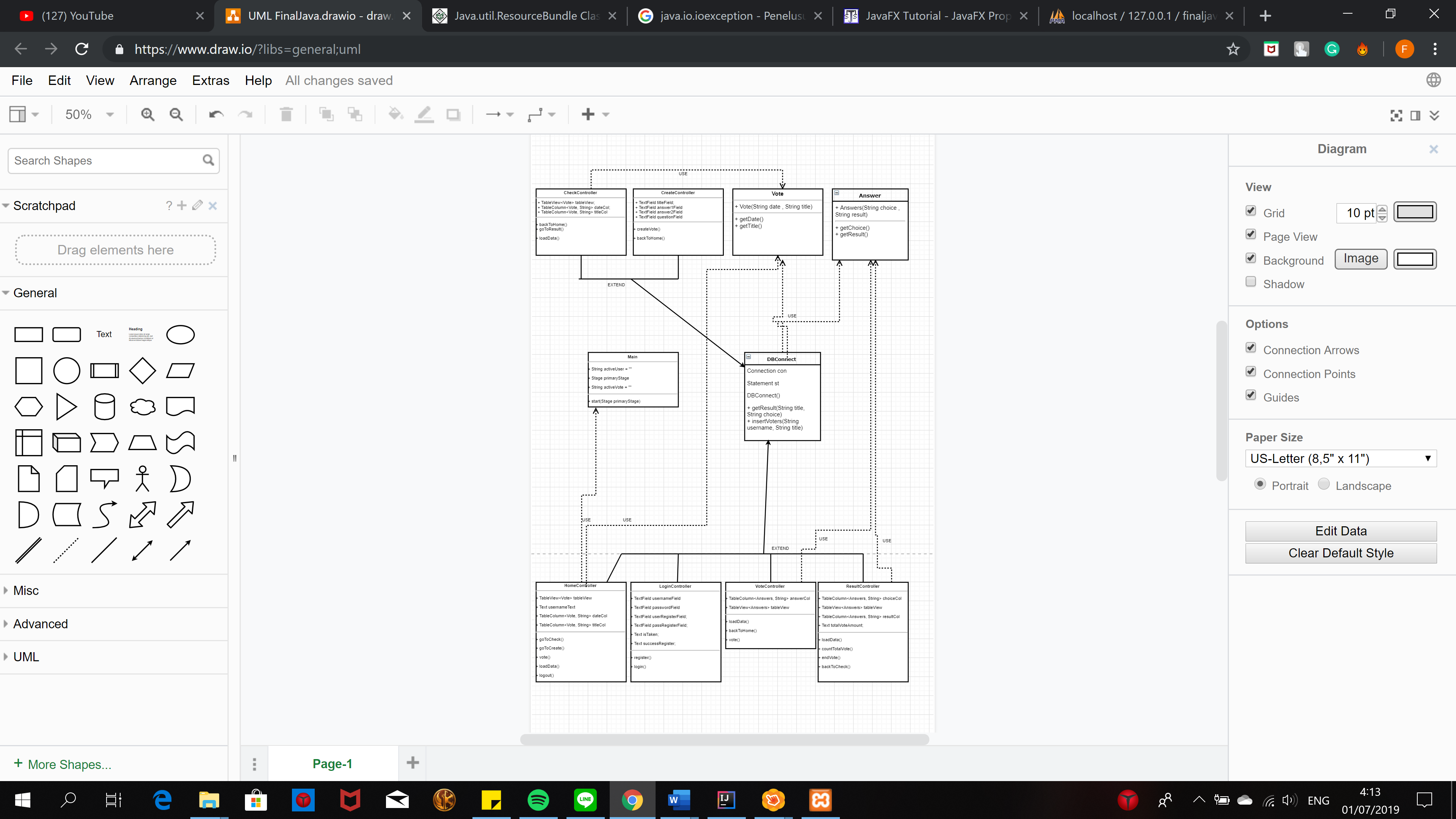
In the existing system, there are no specific websites dedicated to colleges who can participate voting from any location. There are many applications on e-voting for political parties but there are very few web applications which are used for voting or general questions or college-related topics questions.

In the proposed system, we are developing e-voting system for college purpose which will have features of admin and user. Here, the admin will post questions based on social topics or subject related topics related to college. Students can log in and vote to topics and view percentage of votes for each question.

So, for my final project, I have made an application called Vote Anything where all the information can be delivered through one medium. The votes can be categorized by different user with titles of the different kind of information or posts.

For this project, I use mySQL as my database, Java FX with sceneBuilder for my GUI, intelliJ for my IDE and a server.

**IV. Solution Design**



**V. Code Implementation Discussion**

This project imports many different packages from javafx for the GUI, java.time to collect the local current data and format the date, java.sql to connect with a database .

• Coding Language: Java

• Tool: intelliJ

• Database: MYSQL (using XAMPP to connect from localhost/server to mySQL and apache.)

• User Interface: JavaFX with sceneBuilder

import javafx.fxml.FXML; //this import is used to access the fx:id

import javafx.fxml.FXMLLoader; // this import is used to load a new fxml file into the window  
import javafx.scene.Parent;  
import javafx.scene.Scene; // this import is used to set the dimensions of the window

import javafx.stage.Stage; // this import is used to import the stage, so every page will create a new stage

import java.time.format.DateTimeFormatter; // this import is used to format the data dd/mm/yyyy  
import java.time.LocalDate; // this import is used to collect the local current date

import java.sql.\*; // this import is used to connect with the mySQL database in the class DBConnect  
import

import javafx.scene.control.TextField; // this import is used to get the text from the text areas and text fields

import javafx.scene.control.TextArea;

import javafx.application.Application // this import is used to connect with the javafx application in main, to override their start function

**How it works:**

Admin:

Admin can log in with username and password and upload information like questions and two options for each question. Admin can view answers given by students in graphical view in percentage.

User:

The user will log in with another account and view all questions posted by admin and student can vote for all questions.

To understand how the code works, the explanation will be divided into different sections.

1. Login page
2. Home Page
3. Create Vote
4. Vote Page
5. Check Vote
6. Result Page
7. Server
8. **Login page**

The programme starts with the login window, there are two function in this login page. The first is for the user or admin to login using their username and password.

public void login() // create the function to login the registered account

This is the function that will check the username in the database that matches with the one that is typed by the user. If there are none that is a match then the username simply doesn’t exist and it will cease to proceed with the login. If there is a match, then the user will be allowed to move to the home page.

public void register(){ //create the function to register an account

For new users, this is the second function that will allow them to register their username and password into the database. It will then recognised the user if he/she continue to proceed to login to vote.

1. **Home page**

In this programme, each of the user can create their own votes but it also forbid others to see the result other than the creator of the vote.

public void goToCreate() throws IOException { // go to create vote scene

public void logout() throws IOException { // back to login scene

public void goToCheck() throws IOException { // go to check scene

public void vote() throws IOException { // function that allow the user to vote and checked if he/she already voted

There are many options for the user such as to logout, vote, create vote, and show result of the ongoing votes that the user created.

But the main function of this home page is to show the user all of the ongoing vote that are recognised by the system in the database.

public void loadData(){ // get the created data of available votes that exist

try {  
 Connection con = DBConnect.*getConnection*();  
 String sql = "select \* from votes";  
 ResultSet rs = con.createStatement().executeQuery(sql);  
 while (rs.next()) {  
 voteList.add(new Vote(rs.getString("date"), rs.getString("title")));  
 }  
}

1. **Create Vote**

As the name suggest, this window main function is to store a newly created vote and save it in the database while also record the time based on day, month, and year. It will save the new vote under the name of the creator so it will not be possible for others to see the result. There is also the option to cancel this process by clicking the cancel button.

public void createVote() throws IOException { // function that allows the storing of data for the newly created vote

public void backToHome() throws IOException { // move to another scene

1. **Vote Page**

public void loadData(){ // load the answer for the vote

try {  
 Connection con = DBConnect.*getConnection*();  
 String sql = String.*format*("select \* from %s" , Main.*activeVote*);  
 ResultSet rs = con.createStatement().executeQuery(sql);  
 while (rs.next()) {  
 answerList.add(new Answers(rs.getString("choice"), ""));  
 }  
}

Immediately after the user select the vote button in the home page after highlighting one of the selected vote from all the list that is available, this window will show the user two answer based on the loaded data.

public void vote() throws IOException { // function to update amount of voters and result

public void backToHome() throws IOException { // back to home scene

After the user highlight one of the answers, the system will then proceed to update the amount of vote in the database. It will automatically go back to the home page afterwards.

1. **Check Vote**

public void loadData(){ // show the admin created vote based on the name of the creator

try {  
 Connection con = DBConnect.*getConnection*();  
 String sql = String.*format*( "select \* from votes where creator = '%s' ", Main.*activeUser*);  
 ResultSet rs = con.createStatement().executeQuery(sql);  
 while (rs.next()) {  
 voteList.add(new Vote(rs.getString("date"), rs.getString("title")));  
 }  
}

In the home page, if the user clicked the button “show result” then the programme will change scene into another windows that shows the login user all list of votes that he/she have created in the past.

public void backToHome() throws IOException { // back to home scene

public void goToResult() throws IOException { // based on the selected vote, move to result scene to get the result value

The user is then face with choices to go to see the current votes result or go back to the homepage.

1. **Show Result Page**

public void loadData(){ // to Load the percentage based on the current voters

while (rs.next()) {  
 float resultPercent = Float.*parseFloat*(rs.getString("result")) / total \* 100;  
 System.*out*.println(Float.*parseFloat*(rs.getString("result")) / total);  
 String resultInString = rs.getString("result") +" ( " + resultPercent +"%" + " )";  
 answerList.add(new Answers(rs.getString("choice"), resultInString));  
}

The result page is the last window in this programme and its purpose is to show the user the amount of vote on each answer including each percentage and the total amount of voters.

the user also is given choices to either end the ongoing vote or go back to the check vote page.

public void backToCheck() throws IOException { // function to return to another scene

public void endVote() throws IOException { // exterminate current vote function

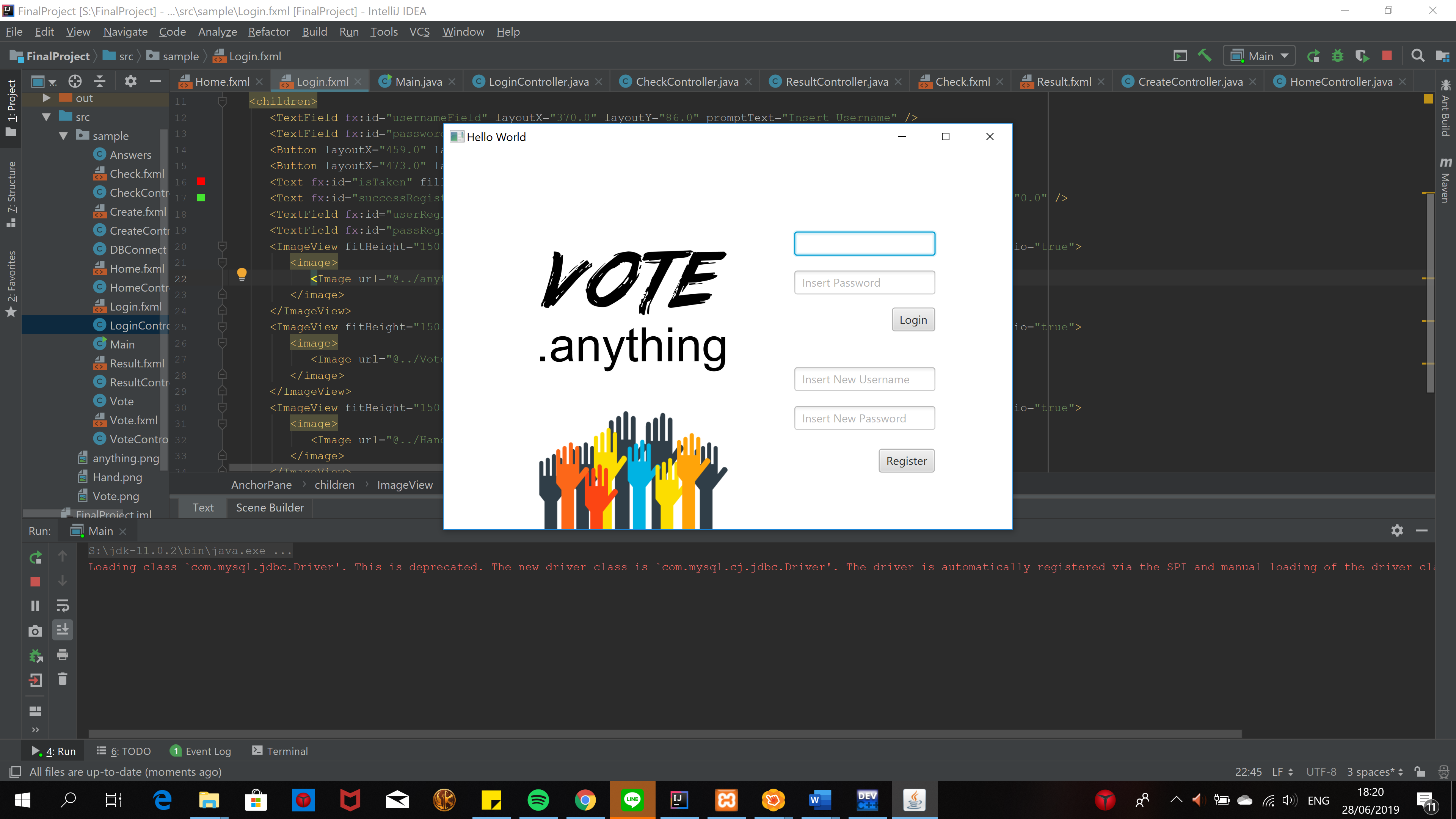
1. **Server**

The database that is used in this programme mySQL uses a software called XAMPP that include apache to open localhost/phpMyAdmin to view the mySQL database.

**VI.** **Evidence of Working Programme**

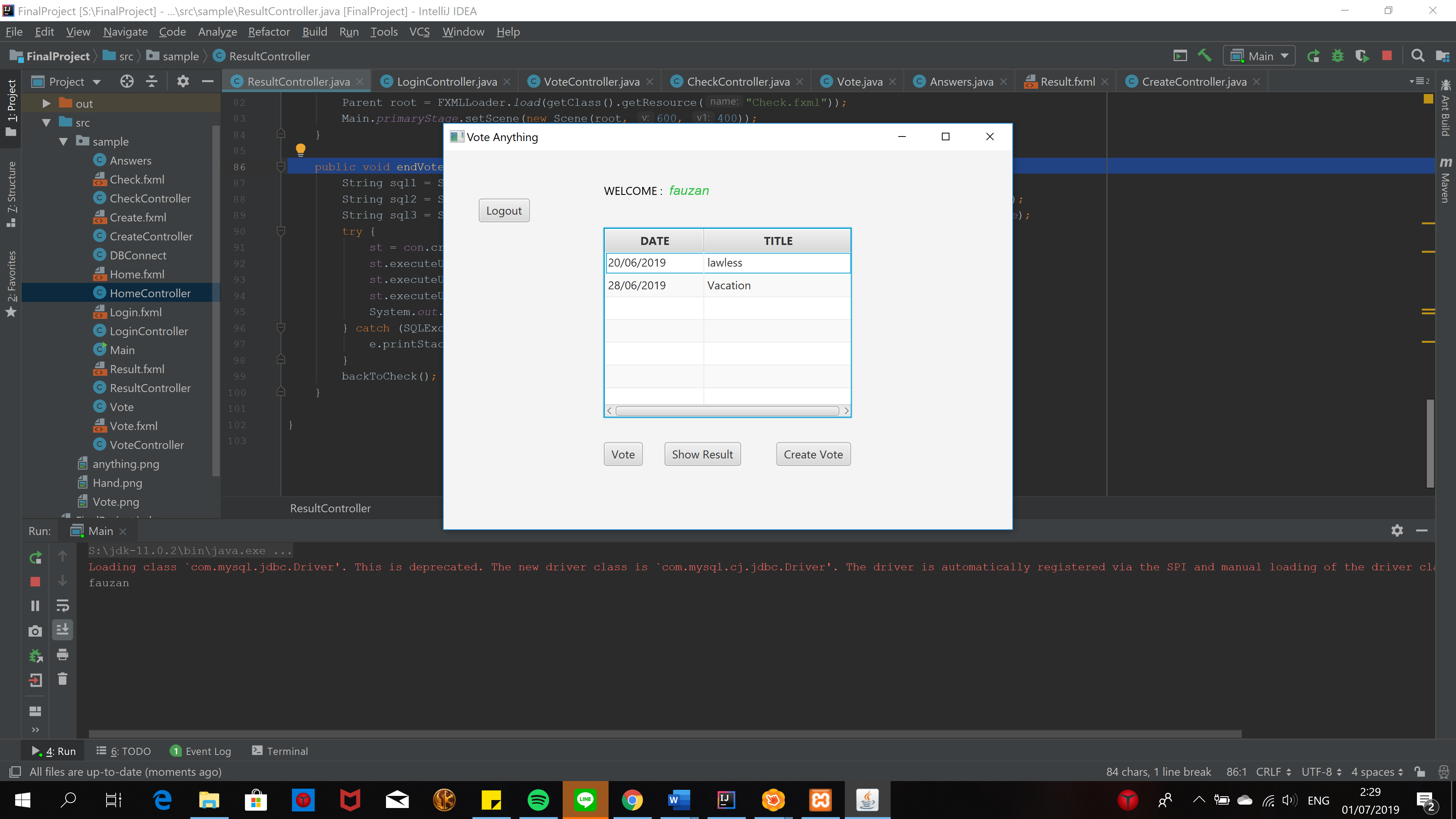
**i. Login page**

This is the login page. When the user launches the application, this window will open. The login page will allow the user/admin to login through this page, see all the current vote list, listed from the most recent to the oldest. Admins can also login through this page. Through logging in, they can get access to admin actions such as: create vote, show result, and end vote.



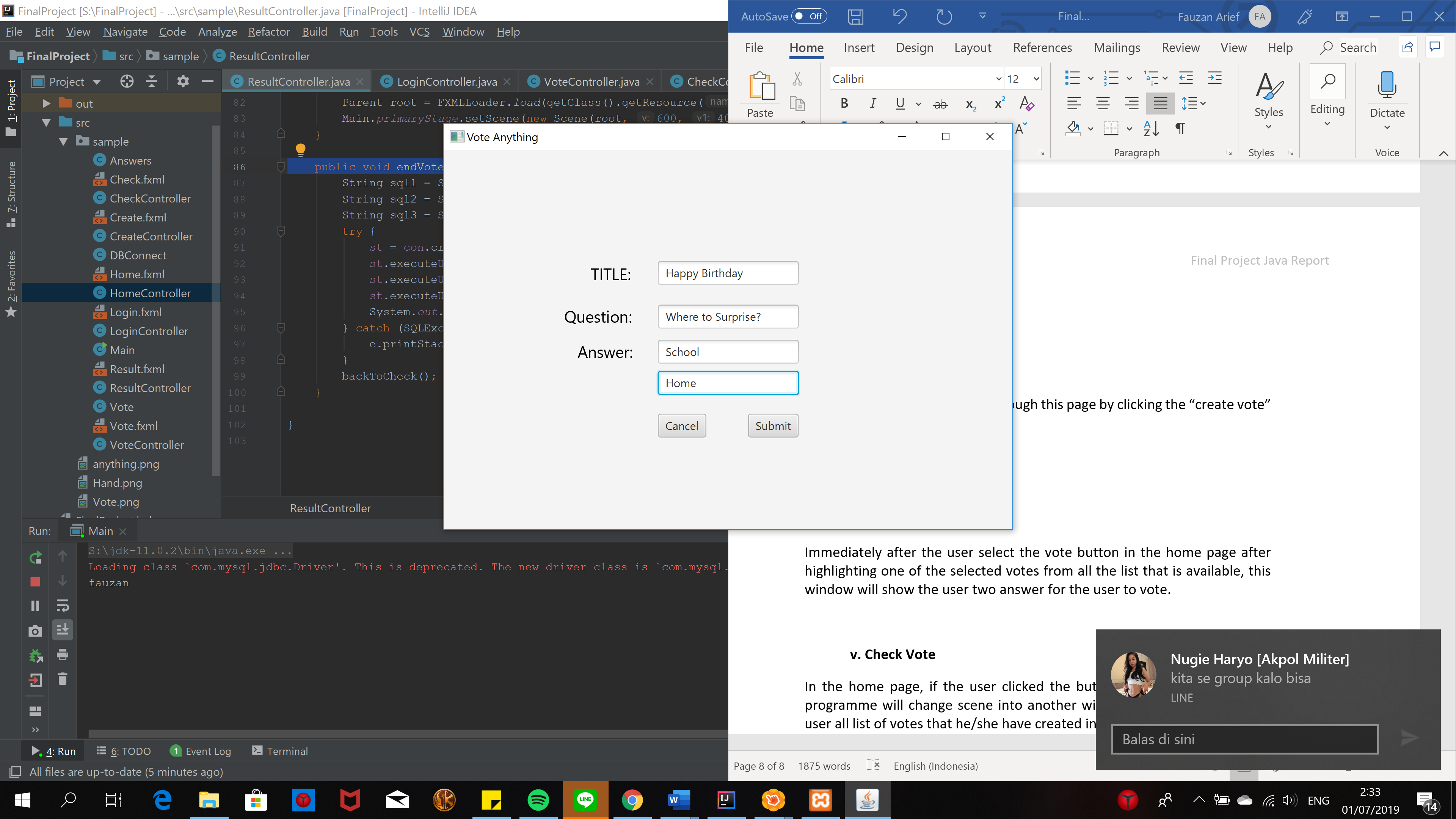
**ii. Home Page**

This window is showing the user all the available vote that are still ongoing.



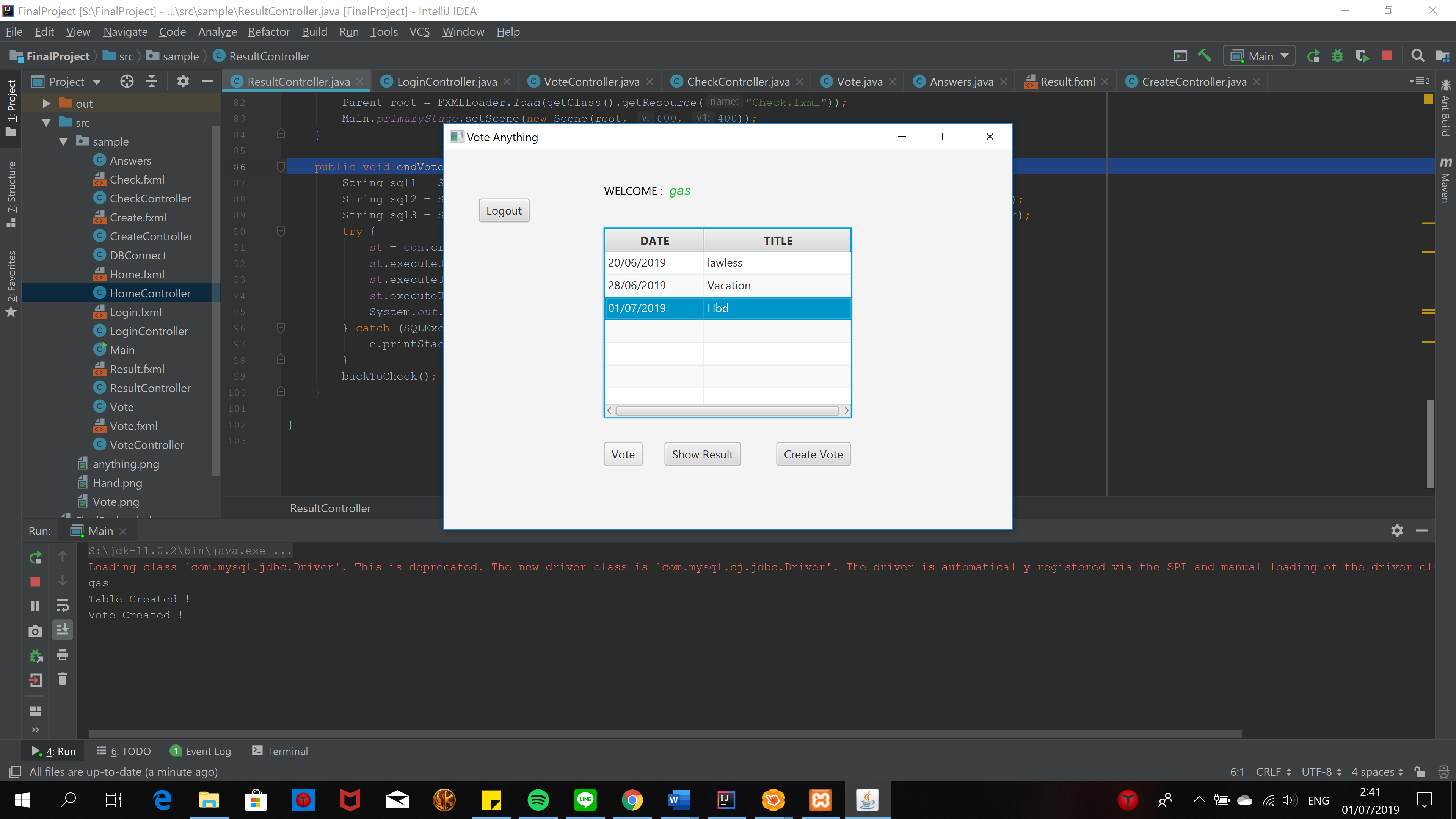
**iii. Create Vote**

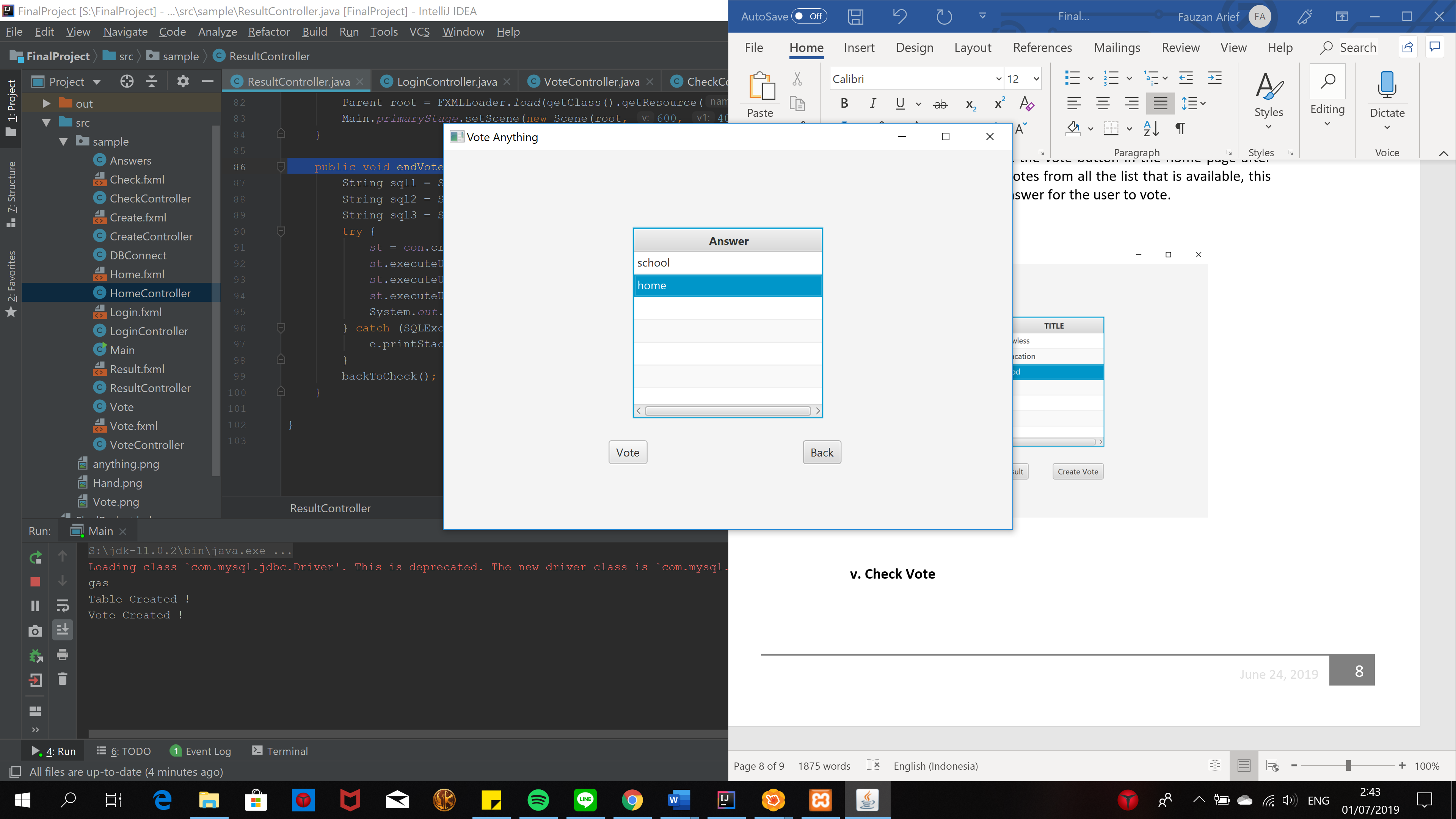
The user can then create vote through this page by clicking the “create vote” button on the previous window.



**iv. Vote Page**

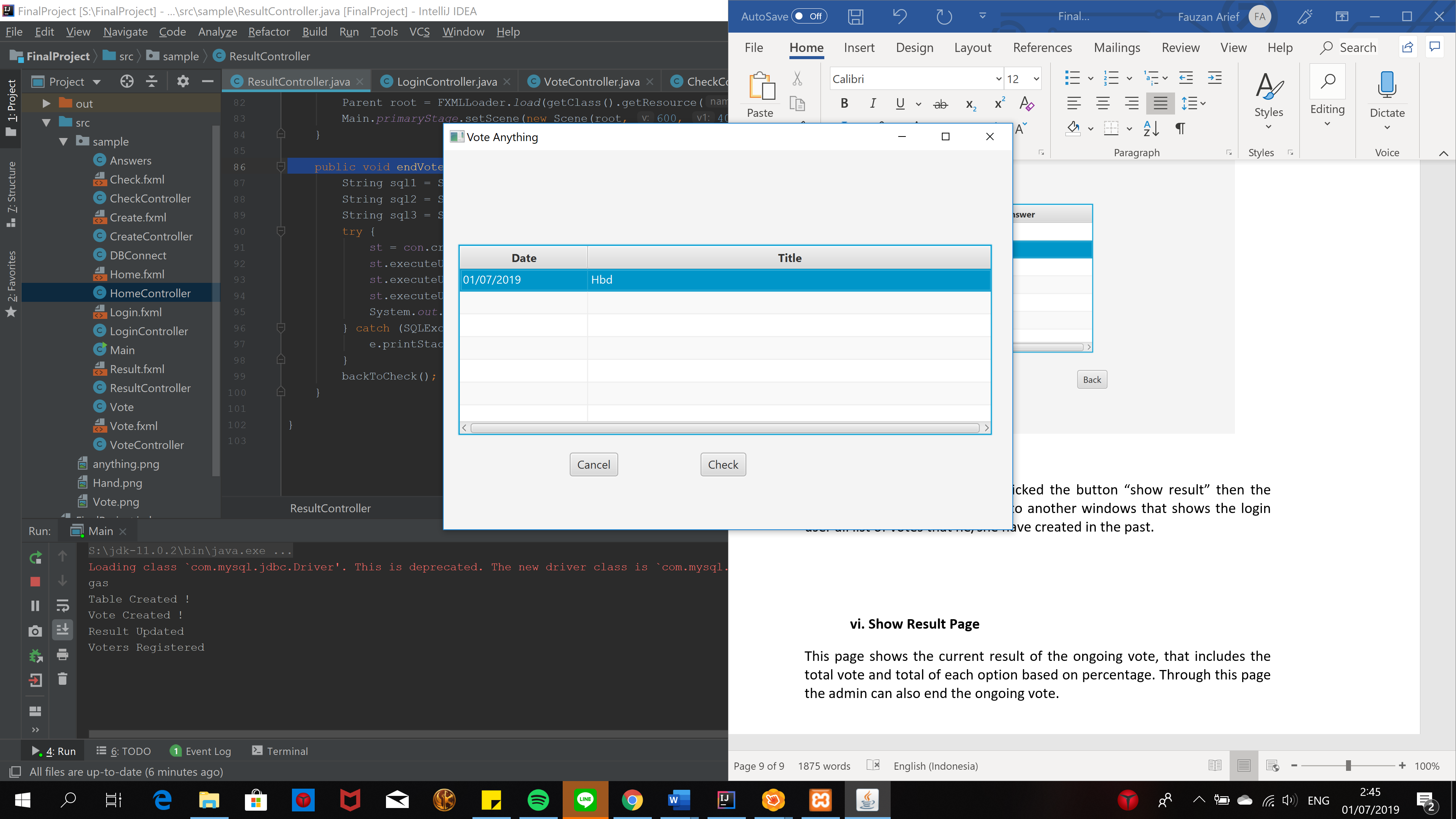
Immediately after the user select the vote button in the home page after highlighting one of the selected votes from all the list that is available, this window will show the user two answer for the user to vote.





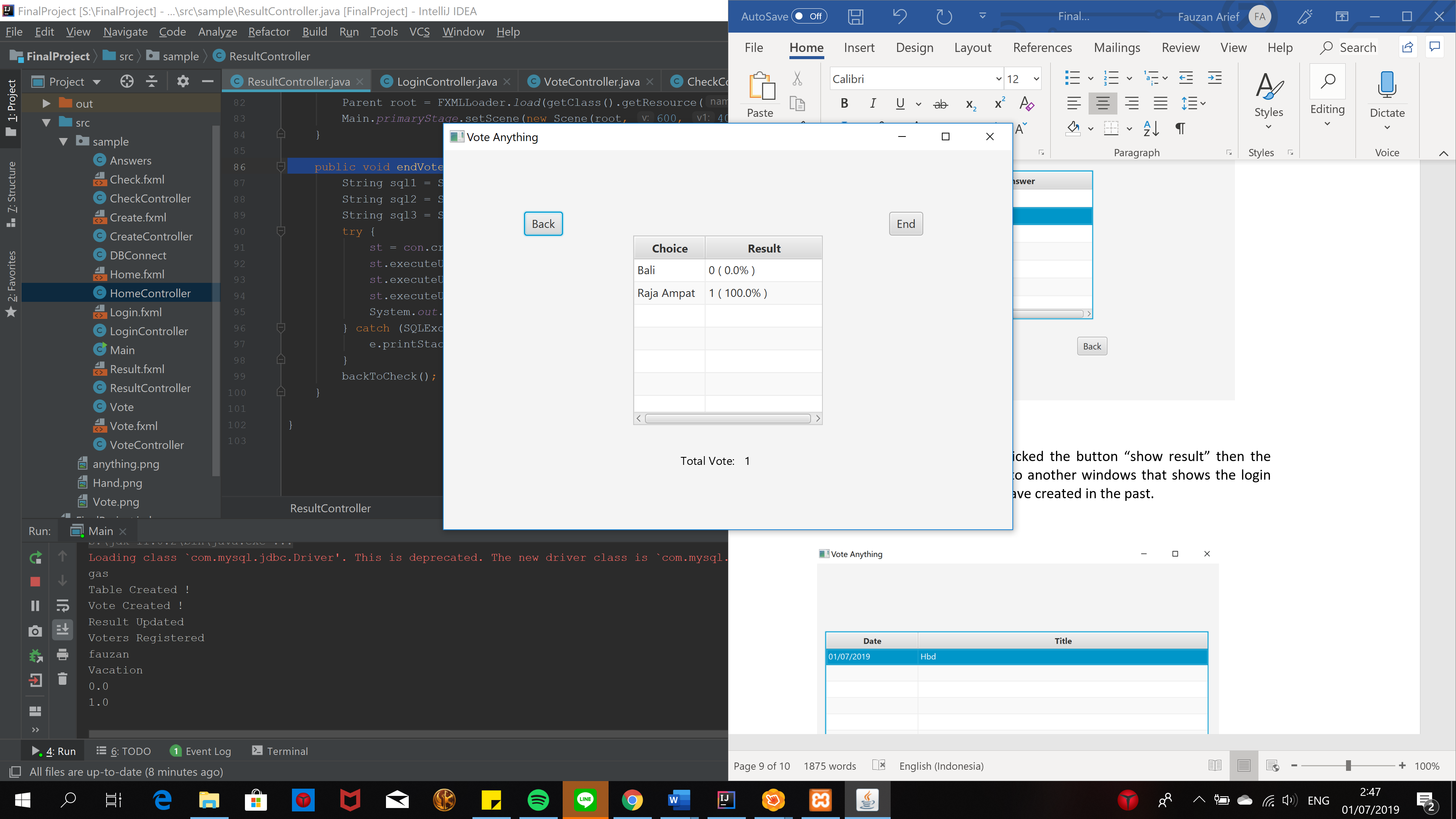
**v. Check Vote**

In the home page, if the user clicked the button “show result” then the programme will change scene into another windows that shows the login user all list of votes that he/she have created in the past.



**vi. Show Result Page**

This page shows the current result of the ongoing vote, that includes the total vote and total of each option based on percentage. Through this page the admin can also end the ongoing vote.



**VII. Credit**

Image:

<https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwicmbL9lZLjAhWBmuYKHRaFChMQjRx6BAgBEAU&url=https%3A%2F%2Fwww.reddit.com%2Fr%2FPewdiepieSubmissions%2Fcomments%2F6t7m5b%2Fanything%2F&psig=AOvVaw3EReAcmF0WSM6WkLwGzbLE&ust=1562017279482132>

<https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwjR6viulpLjAhUZ4XMBHbsaDQoQjRx6BAgBEAU&url=https%3A%2F%2Fdlpng.com%2Fpng%2F195569&psig=AOvVaw3Fk0f4pFrSoT8tdCqDzFbh&ust=1562017384708281>

<https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwjCvqi_lpLjAhXt6XMBHRERBAsQjRx6BAgBEAU&url=https%3A%2F%2Fwww.kisspng.com%2Fpng-voting-display-resolution-clip-art-vote-transparen-422206%2F&psig=AOvVaw3qps_Zo6pTstLiuIn70jiT&ust=1562017423031191>