# **Group Project - Deliverable 3**

## **TEAM – DSEs**

# User Manual:

The user must see the home page as below up on accessing our web application. It gives the options to ‘Join’ or ‘Login’ into our application

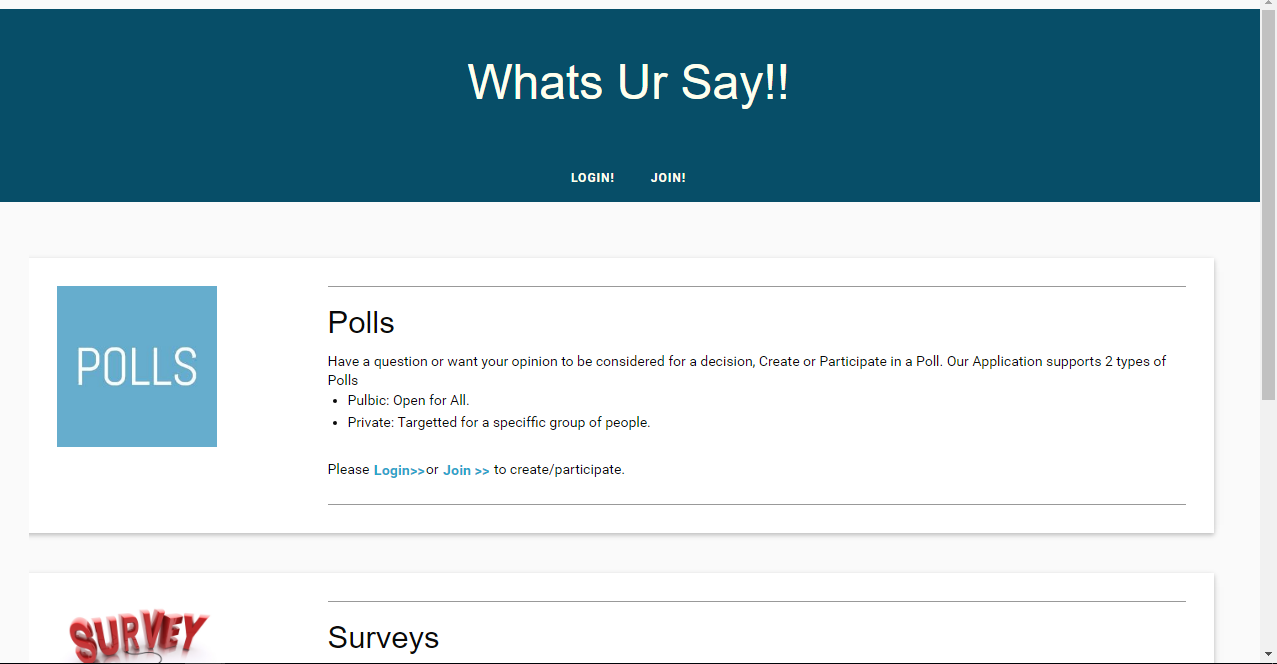


Figure 1: Home Page

If you are the first time user, click ‘JOIN!’ to register, else click ‘LOGIN!’ to login into the system

Click on JOIN! Link page will redirect to registration page where you provide all details as shown in below image to register in the system.

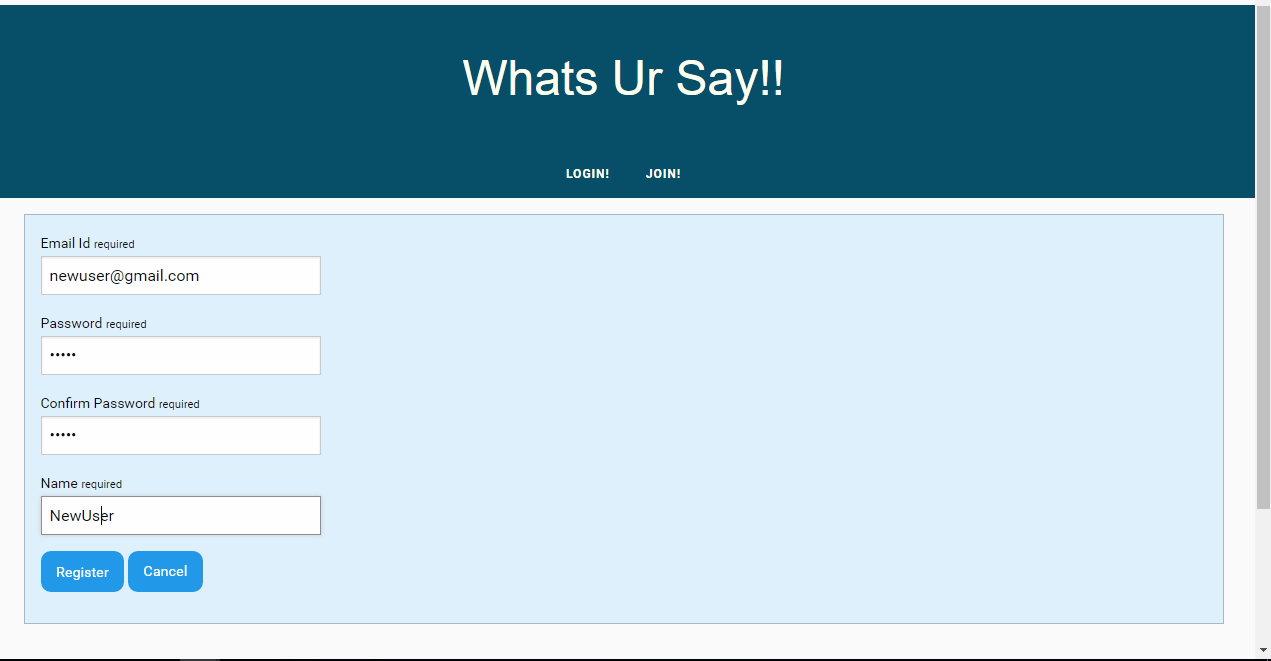


Figure 2: Register Page

Provide all fields’ values and click register. Below popup will be displayed on successful registration and click ok page is redirected to login page.

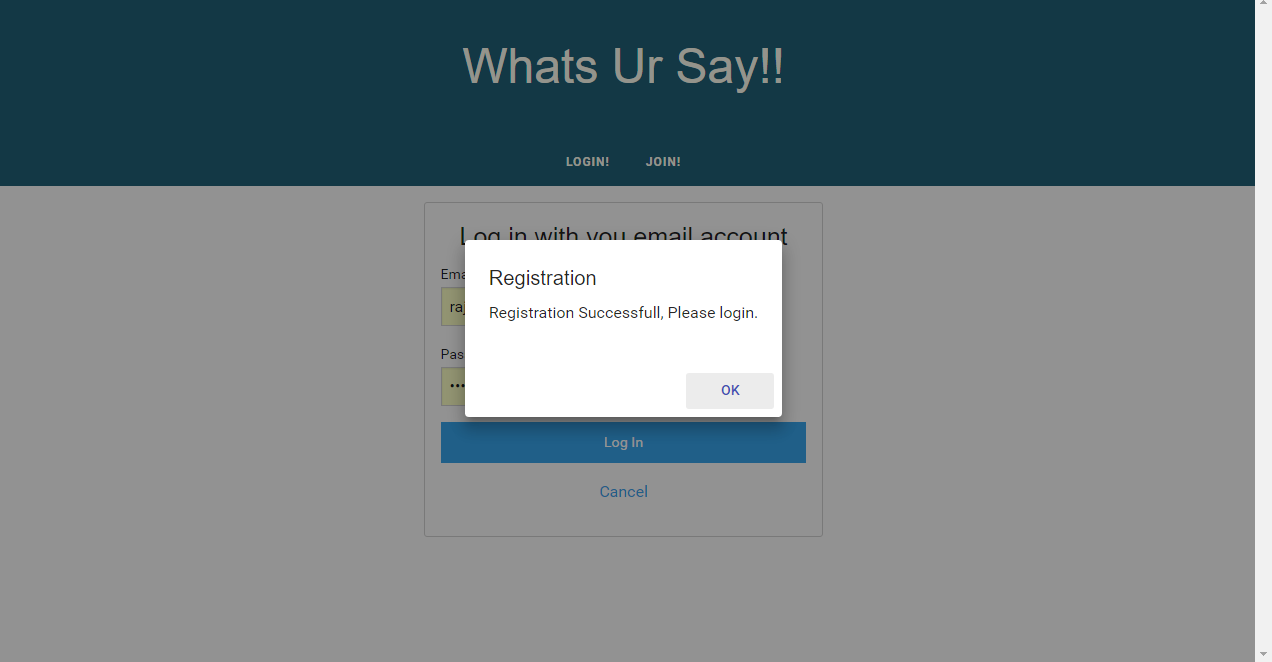


Figure 3: Registration Successful page

To login, provide email and password that is entered during registration as shown in below image and click Login.

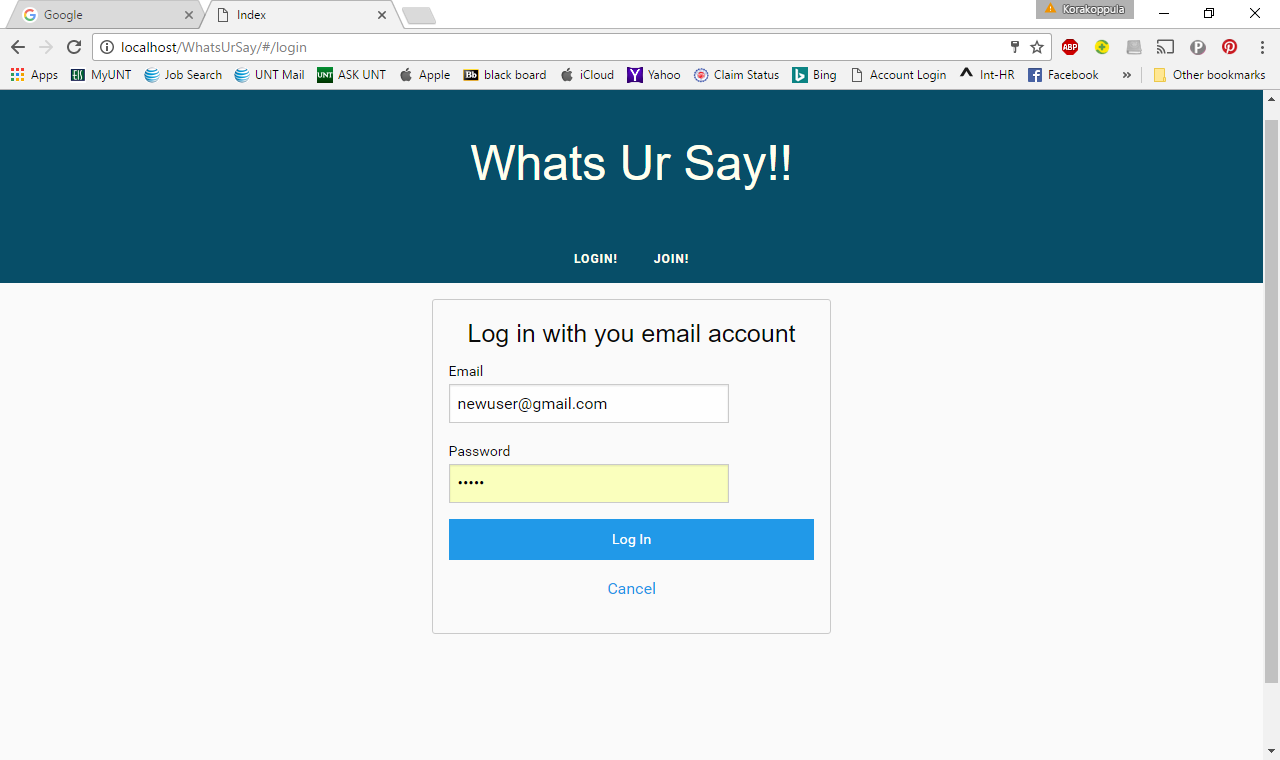
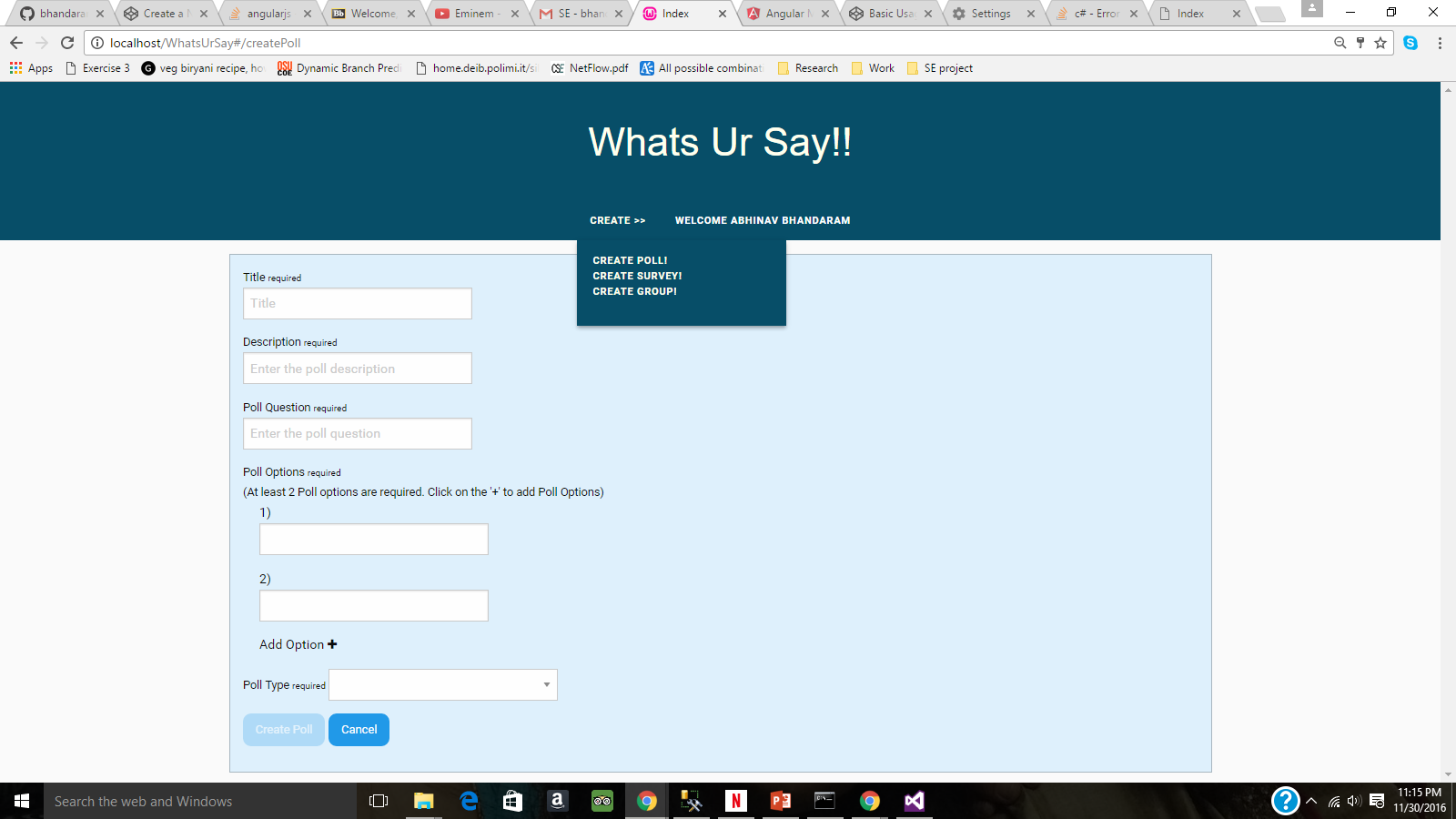


Figure 4: Login Page

**Options to create Poll, Survey, Group:**

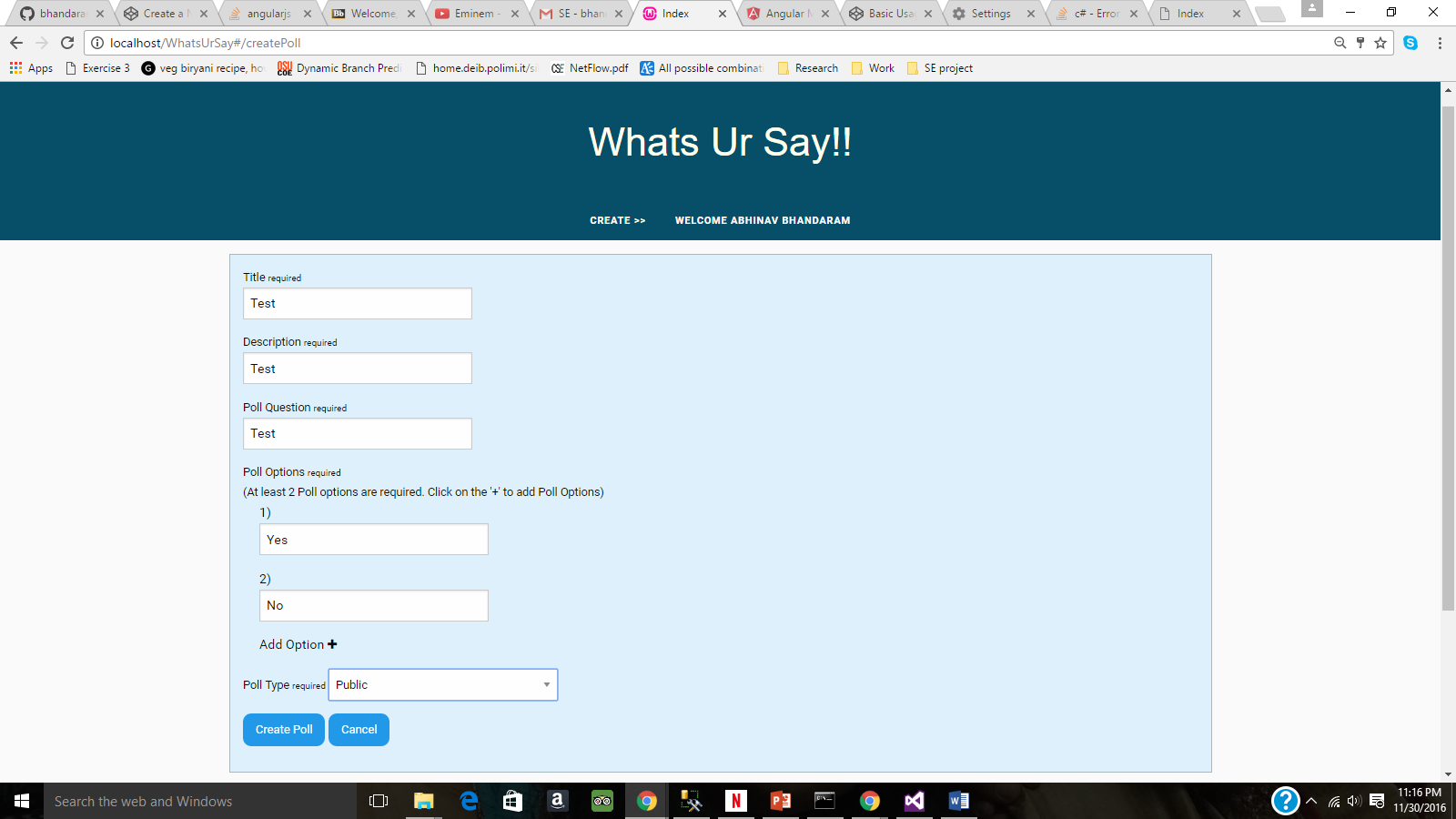
As an admin, clicking on ‘create’ gives options to create poll, survey and group



**Figure 5: Page showing options to create poll, survey and group**

**Create Poll:**

Clicking on ‘Create Poll’ displays the poll creation page as below



**Figure 6: Create Poll**

**Participate in Poll:**

On login dashboard page is shown as below. If you are part of any group and group has any polls associated with it then PRIVATE tab under POLLS will display the polls to participate.

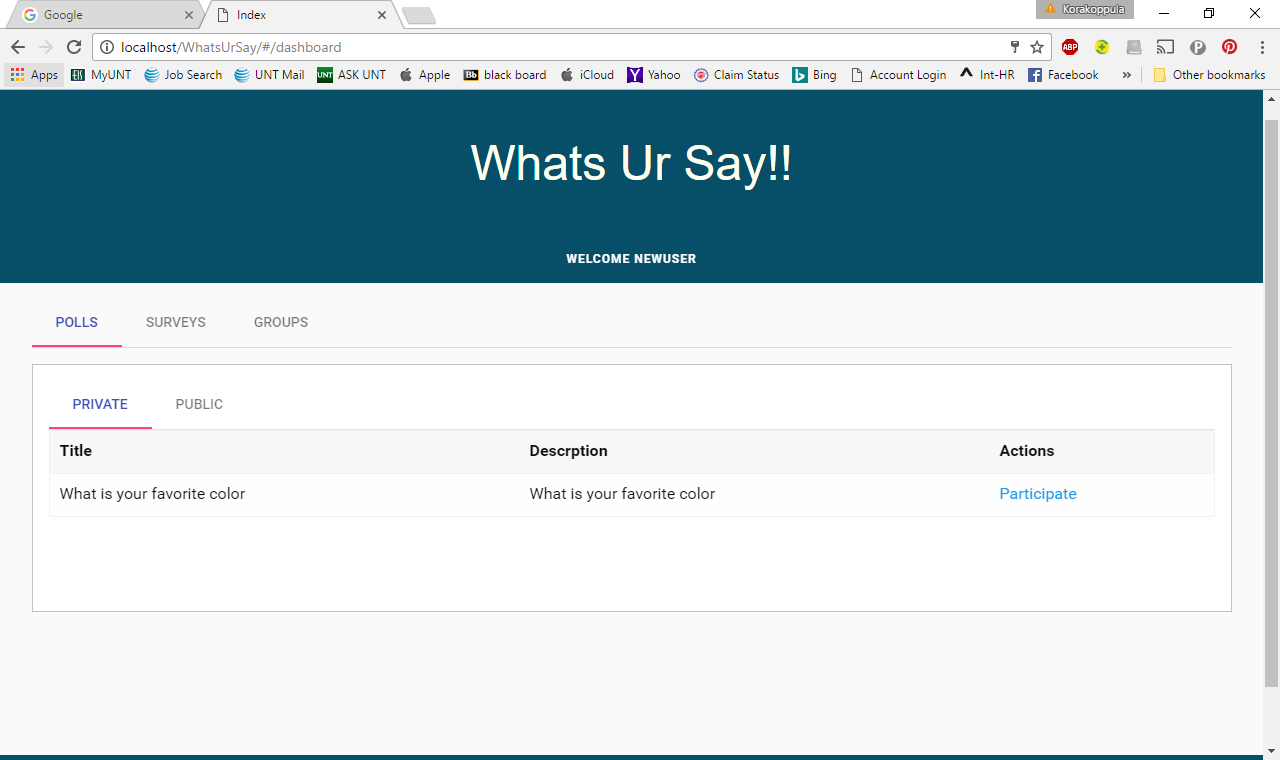


Figure 7: View polls ‘private’ and ‘public’ page

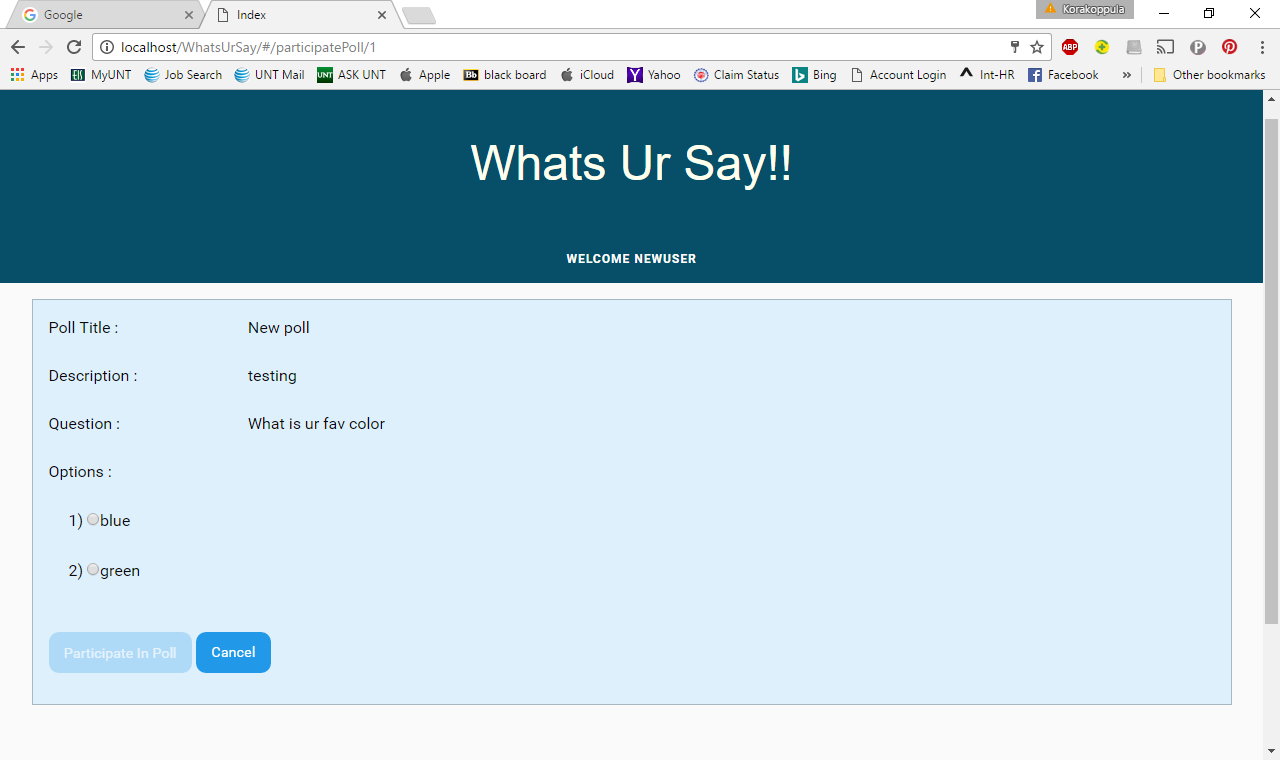
If poll is available click on participate link to participate in poll. You will be redirected to participate page as shown below. Select an option and click to particpate in poll. 

Figure 8: Participate in poll

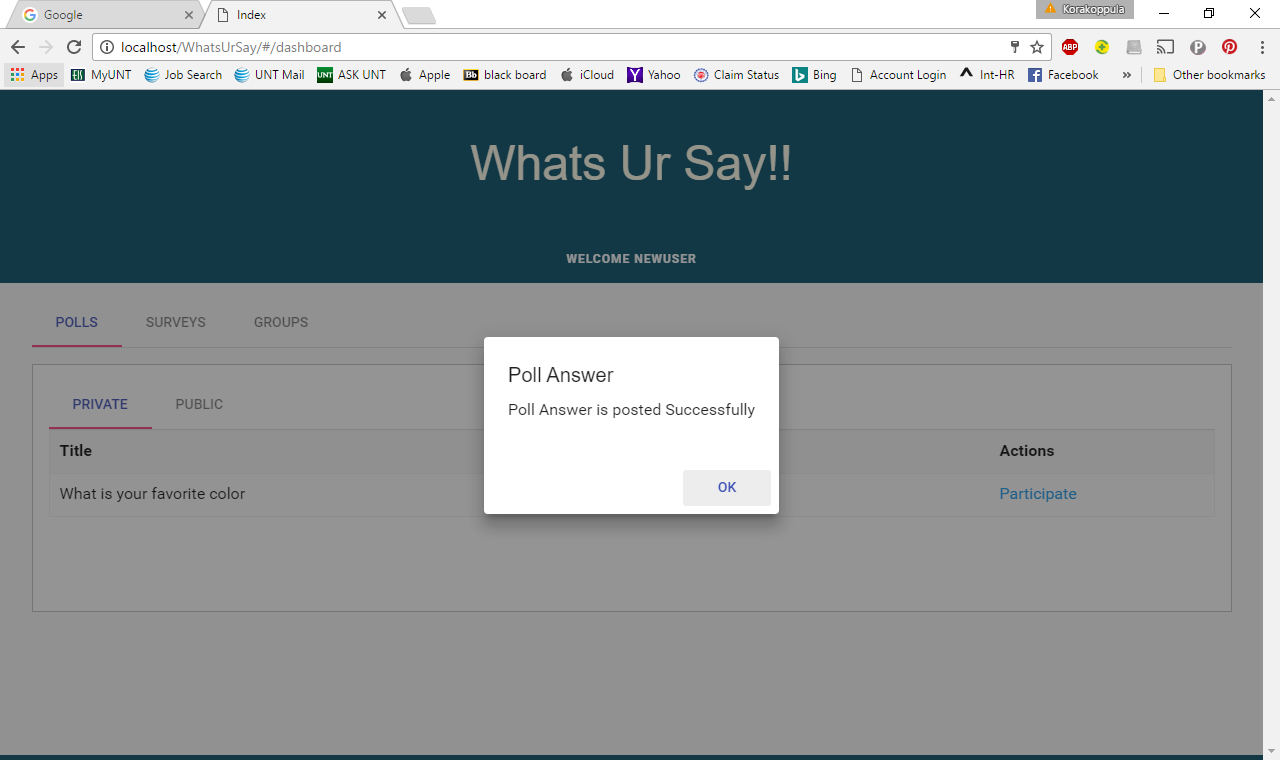


Figure 9: Successful participation in poll

6) PUBLIC tab under POLLS will display all polls which are not related to any group.

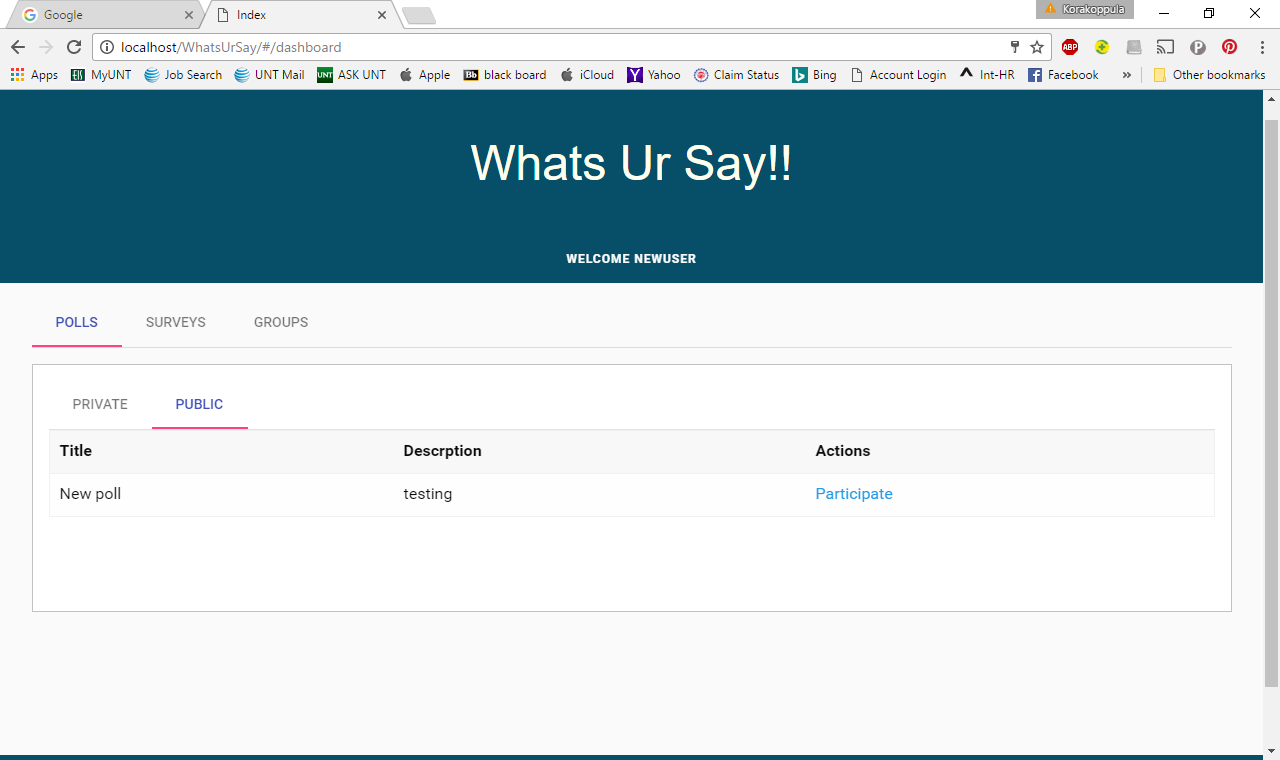
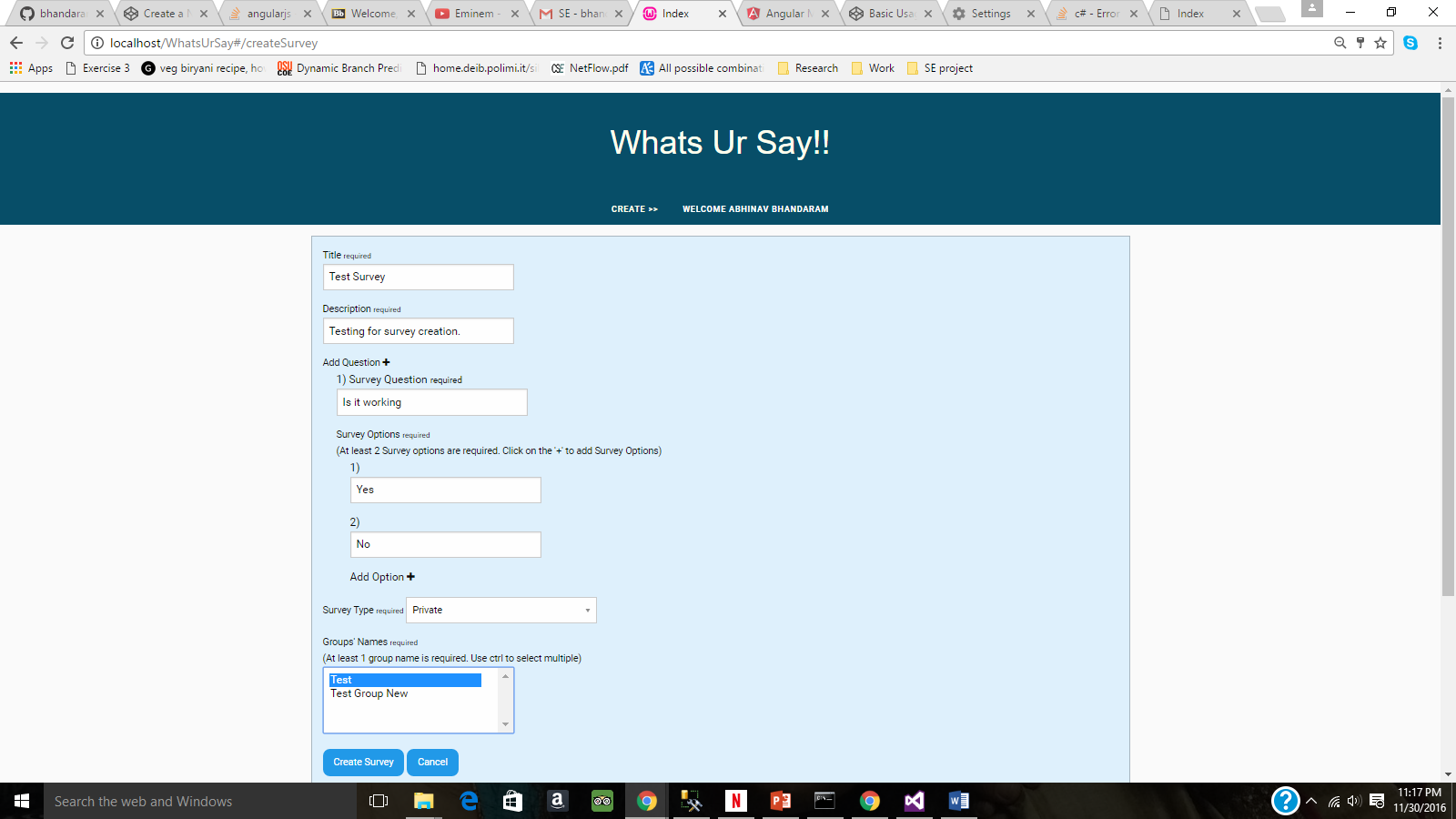


Figure 10: Options to view ‘Public’ and ‘Private’ polls

**Create Survey:**

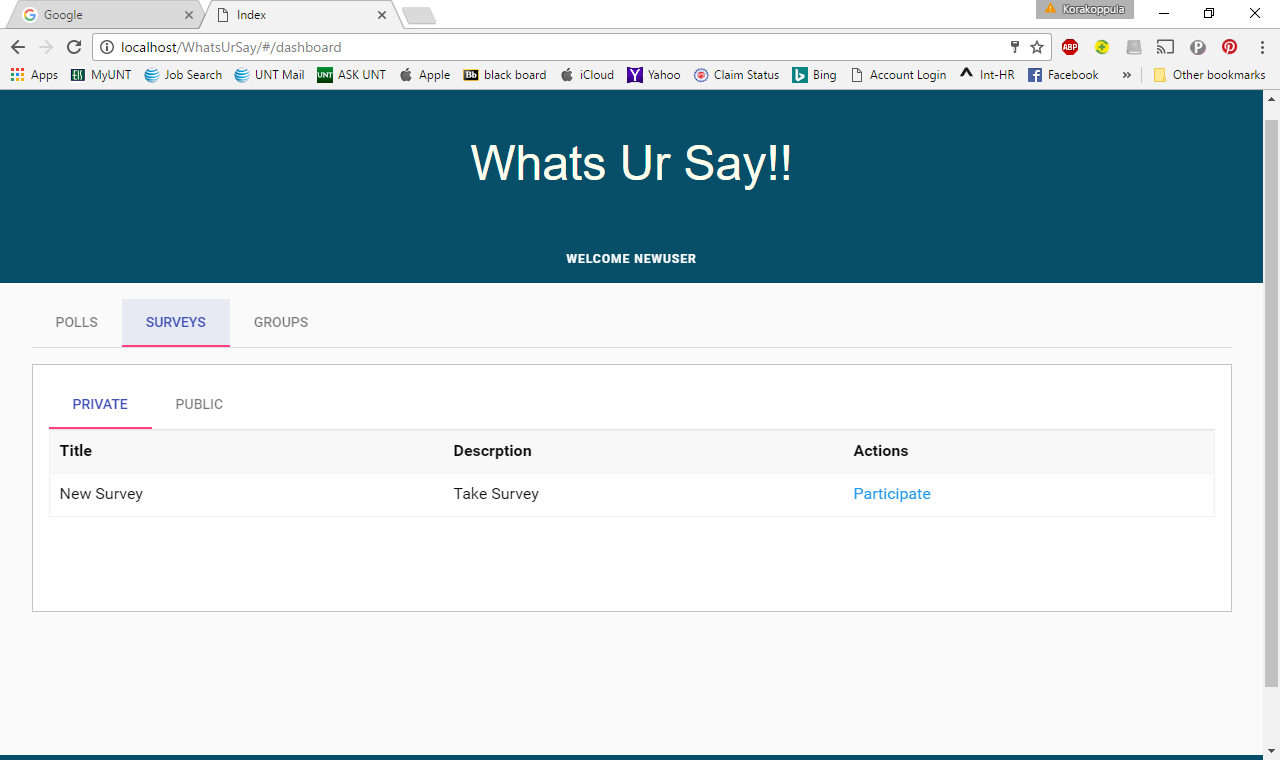
Clicking on ‘Create Survey’ under ‘Create’ button will result in the survey creation page as below



**Figure 11: Create Survey**

**Participate in Survey:**

Navigate to Survey tab. If you are part of any group and group has any survey associated with it then PRIVATE tab under SURVEYS will display the surveys to participate.



**Figure 12: Option to view the polls in which the user can participate in**

If survey is available click on participate link to participate in survey. You will be redirected to participate page as shown below. Select an option for each question and click Submit Survey.

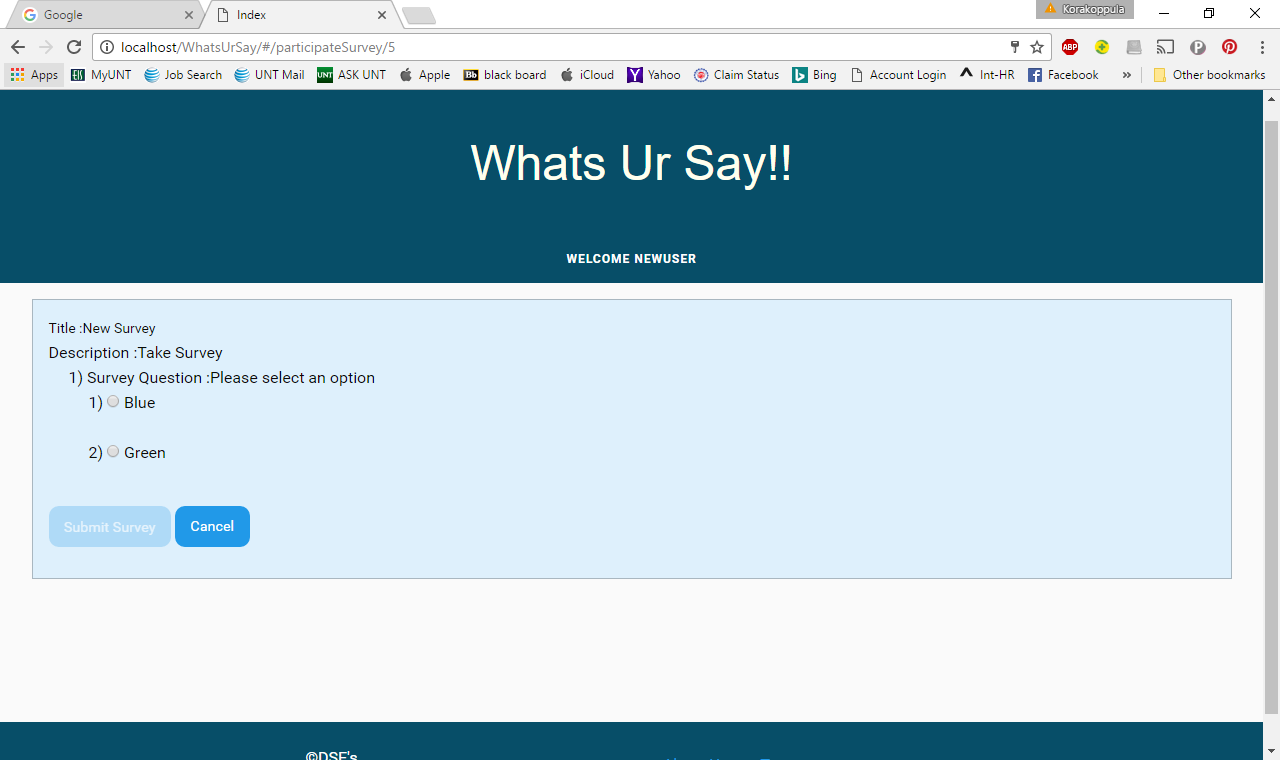


Figure 13: Participate in Survey

**View Poll Results:**

Clicking on ‘View Results’ will give the results page to the users, which contains the results of a poll

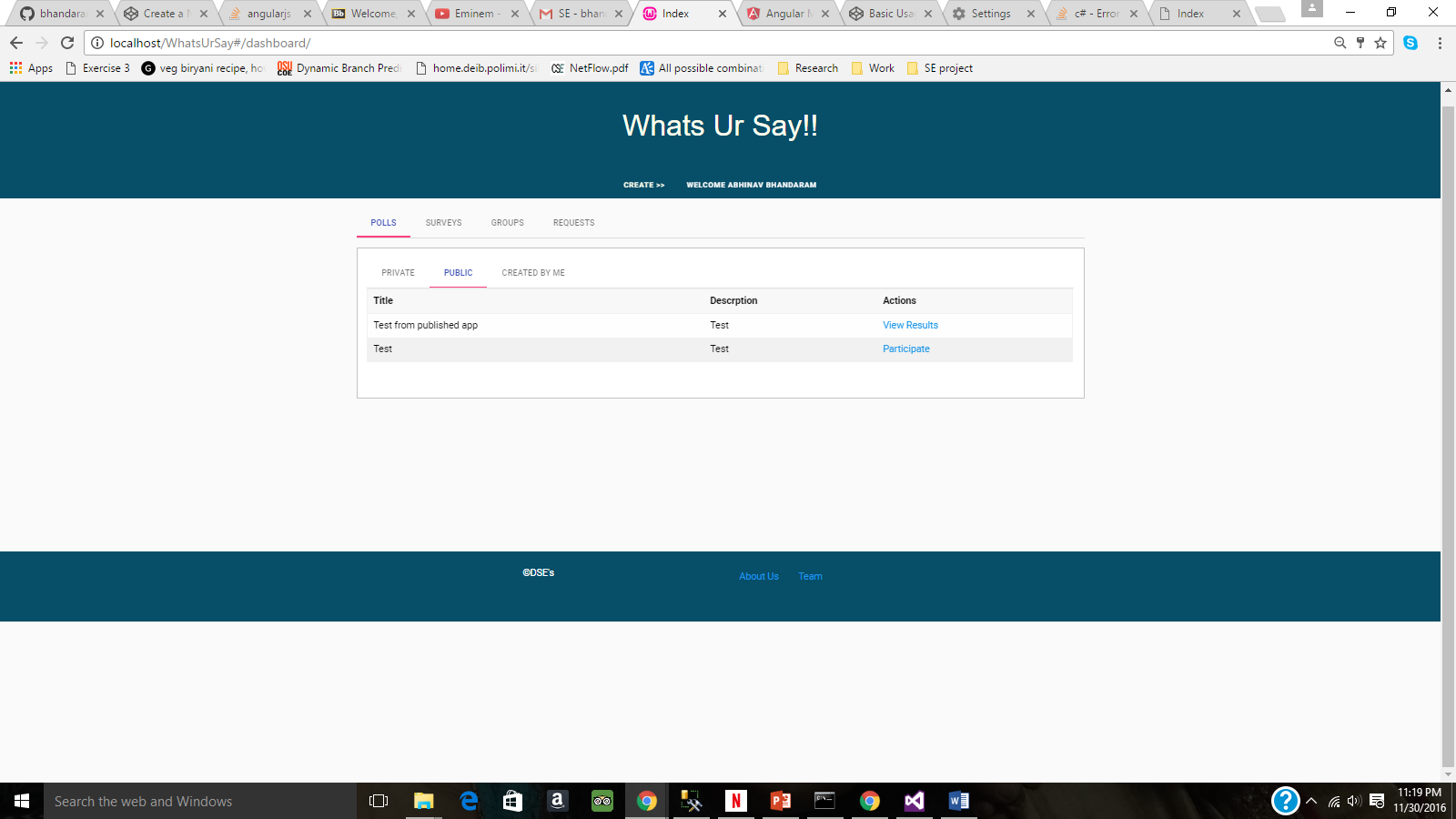


Figure 14: Page displaying the list of public polls

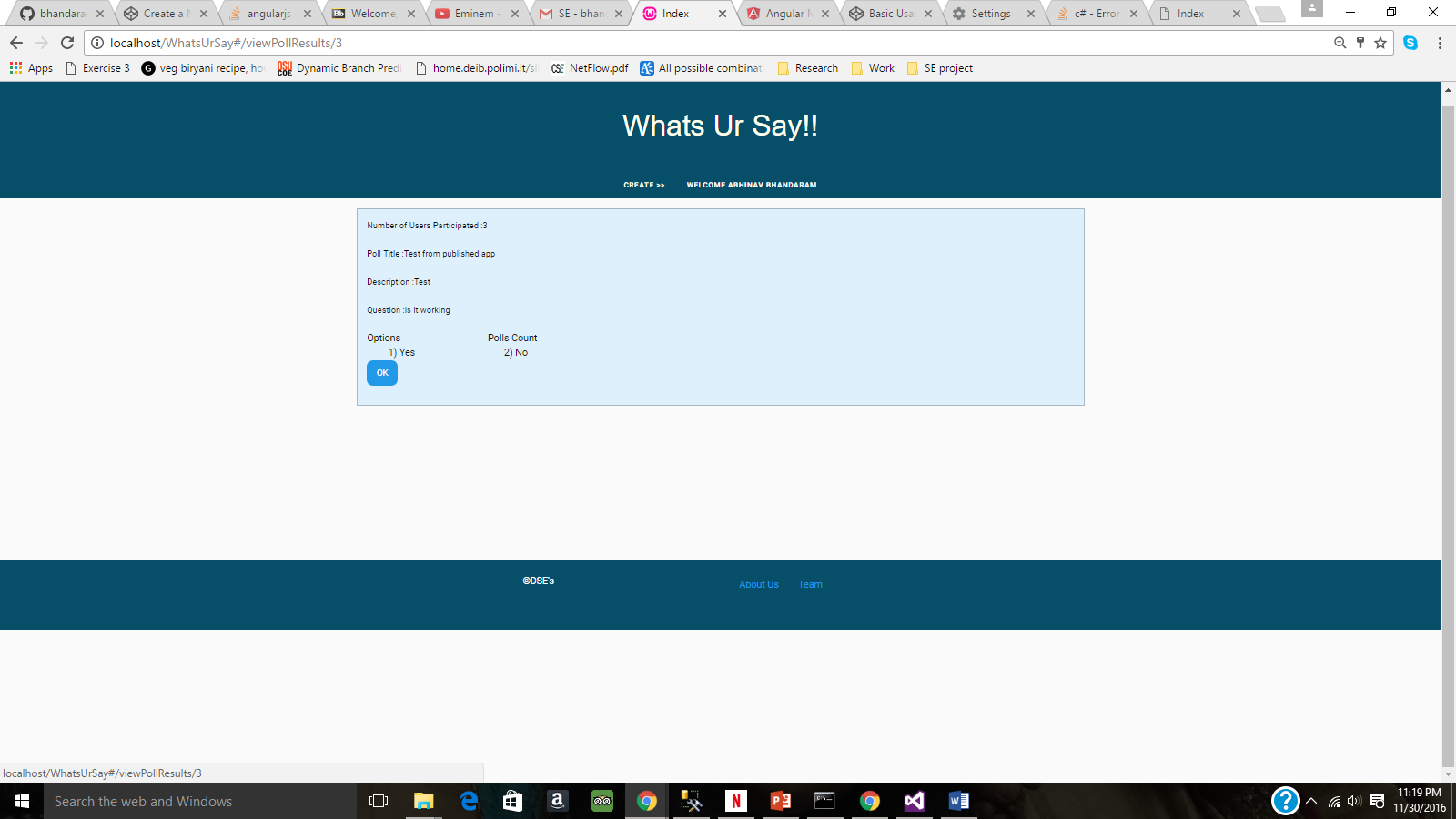


Figure 15: View poll results

**Create Group:**

Clicking on ‘Create Group’ under ‘create’ button gives a page as below, where the user can create a user group

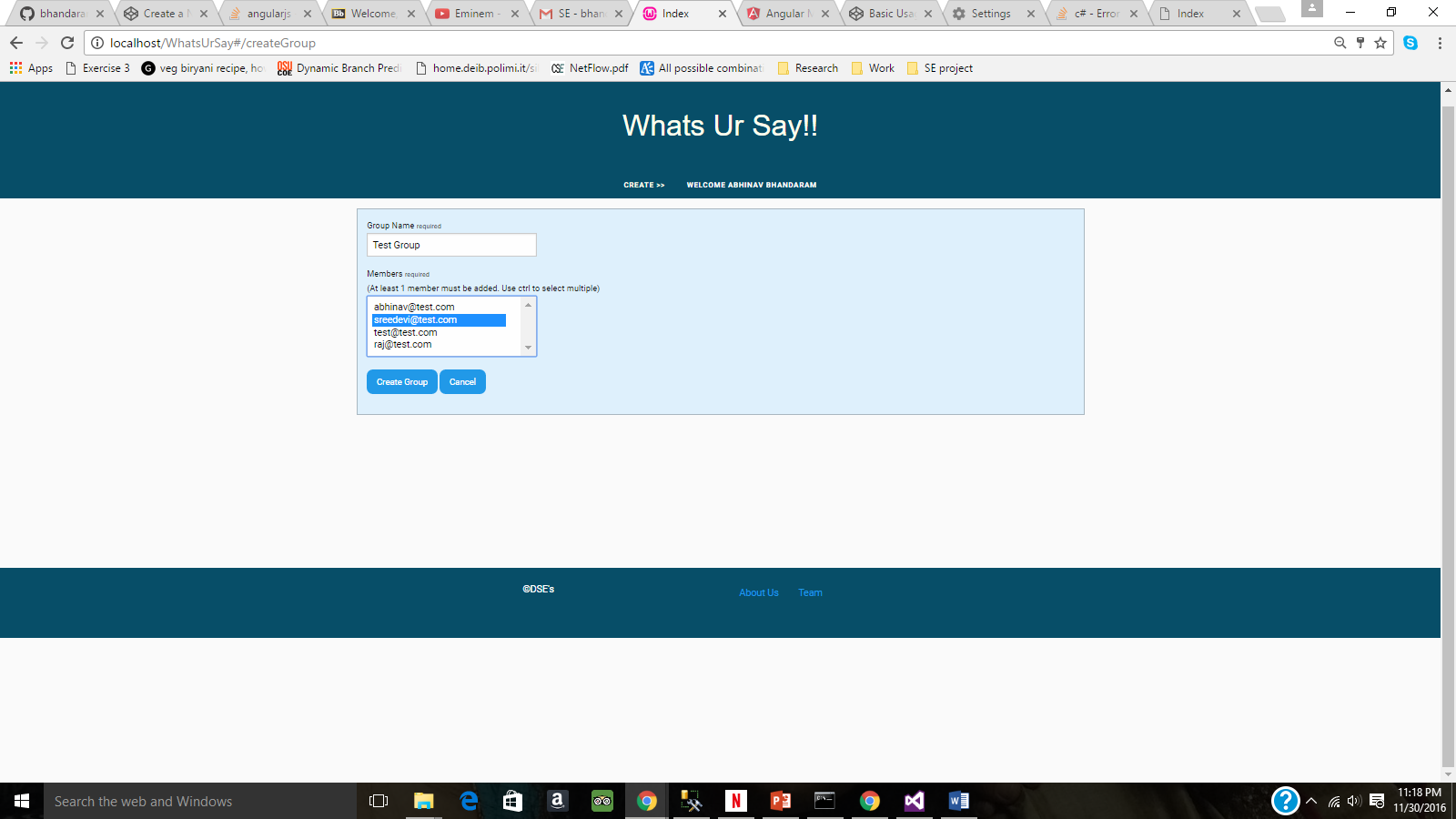
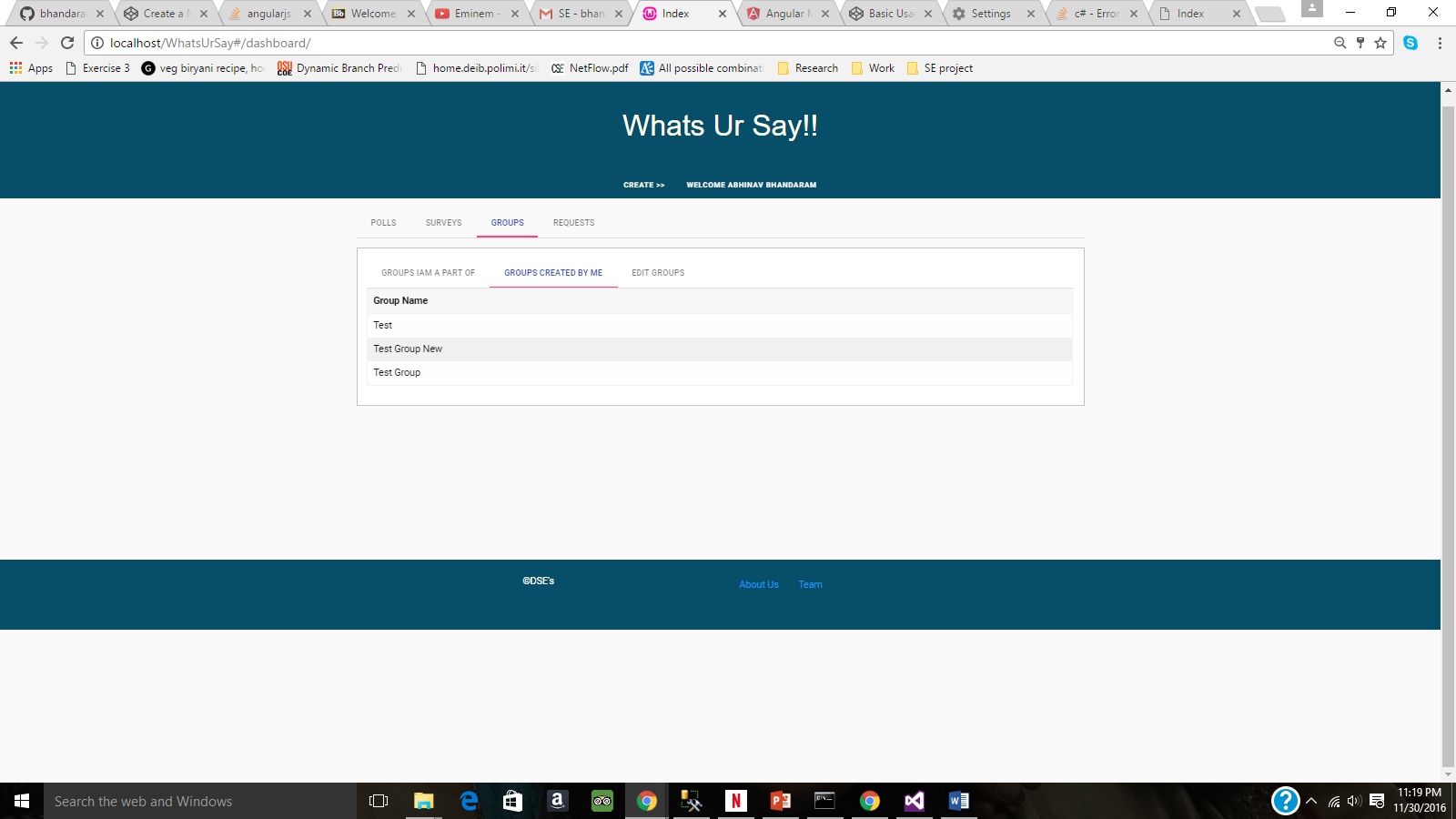


Figure 16: Group Creation page

**Option to view the Groups related to a user:**

The user dashboard contains options to view the group details, where a user can view the groups that he/she is a part of and the groups that are created by that particular user



**Figure 17: Page displaying list of groups created by a user or**

**List of groups where the user is a part of.**

**Edit Group:**

The dashboard of an authorized user has an option to edit a user group, where any number of members can be added/removed to or from the group

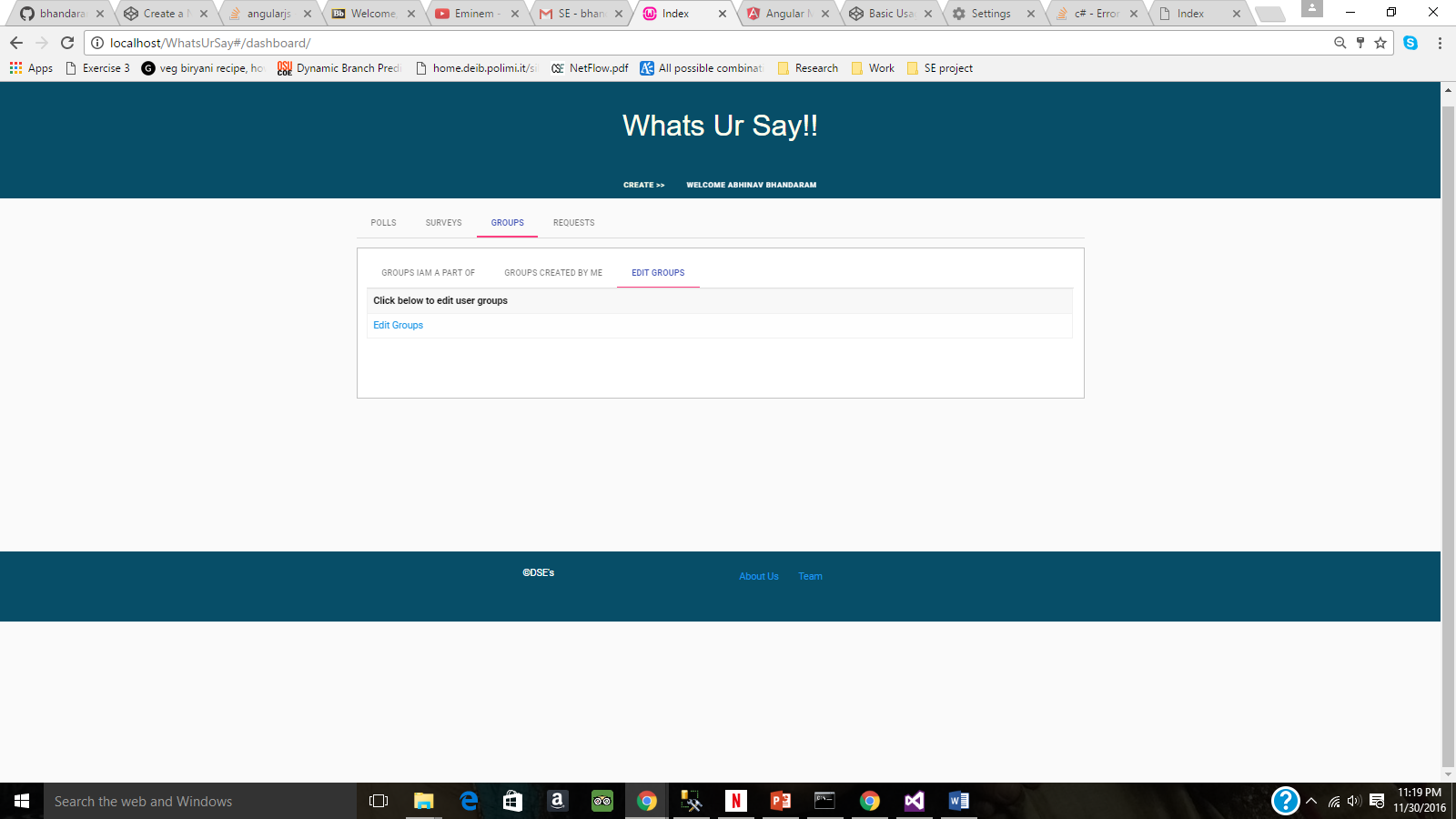


Figure 18: Dashboard displaying the option to ‘Edit a Group’

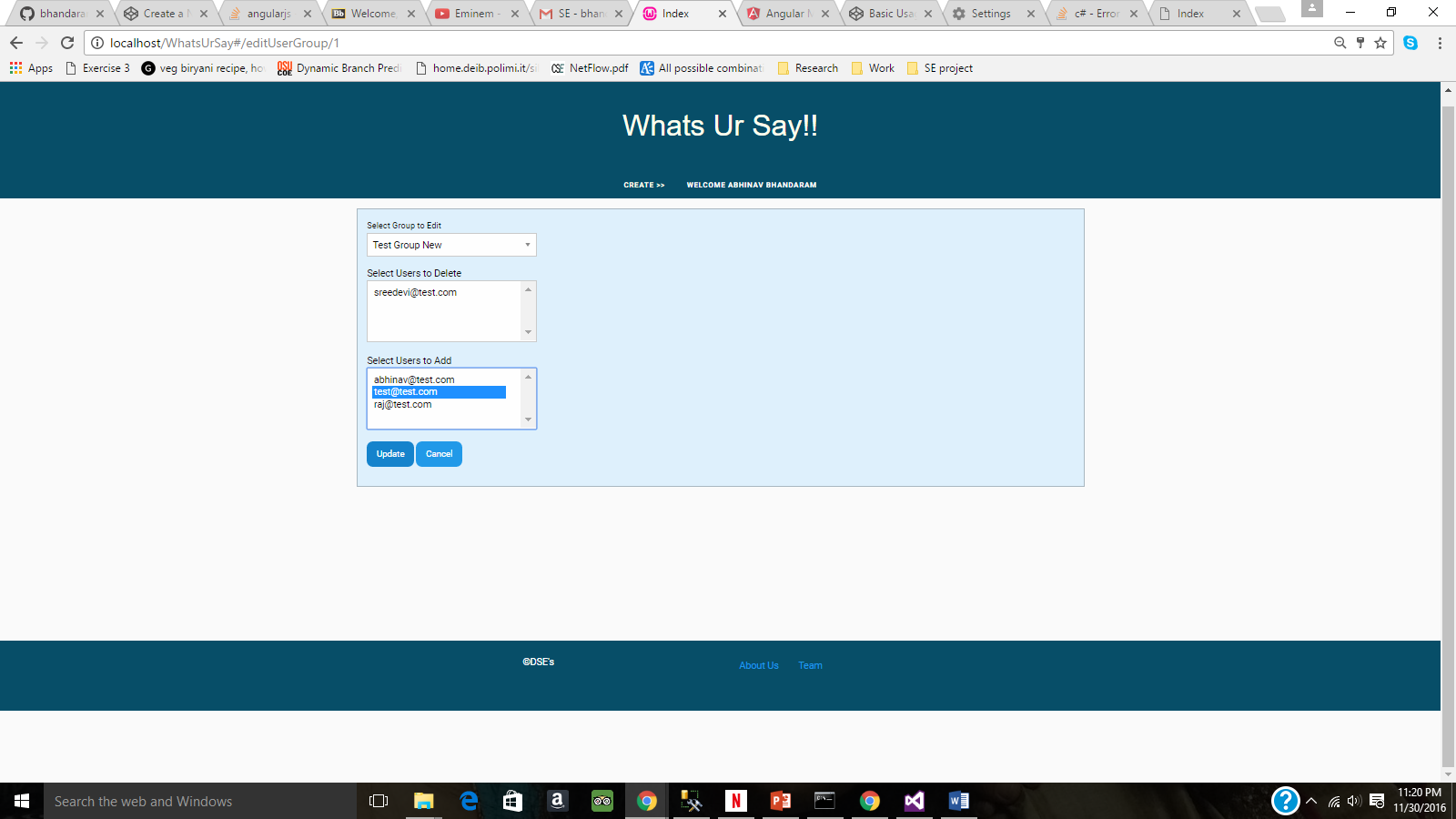


Figure 19: ‘Edit Group’ page

**Request for Group Leadership:**

The user can request for the role of group leadership by clicking the ‘Request Group Leadership’

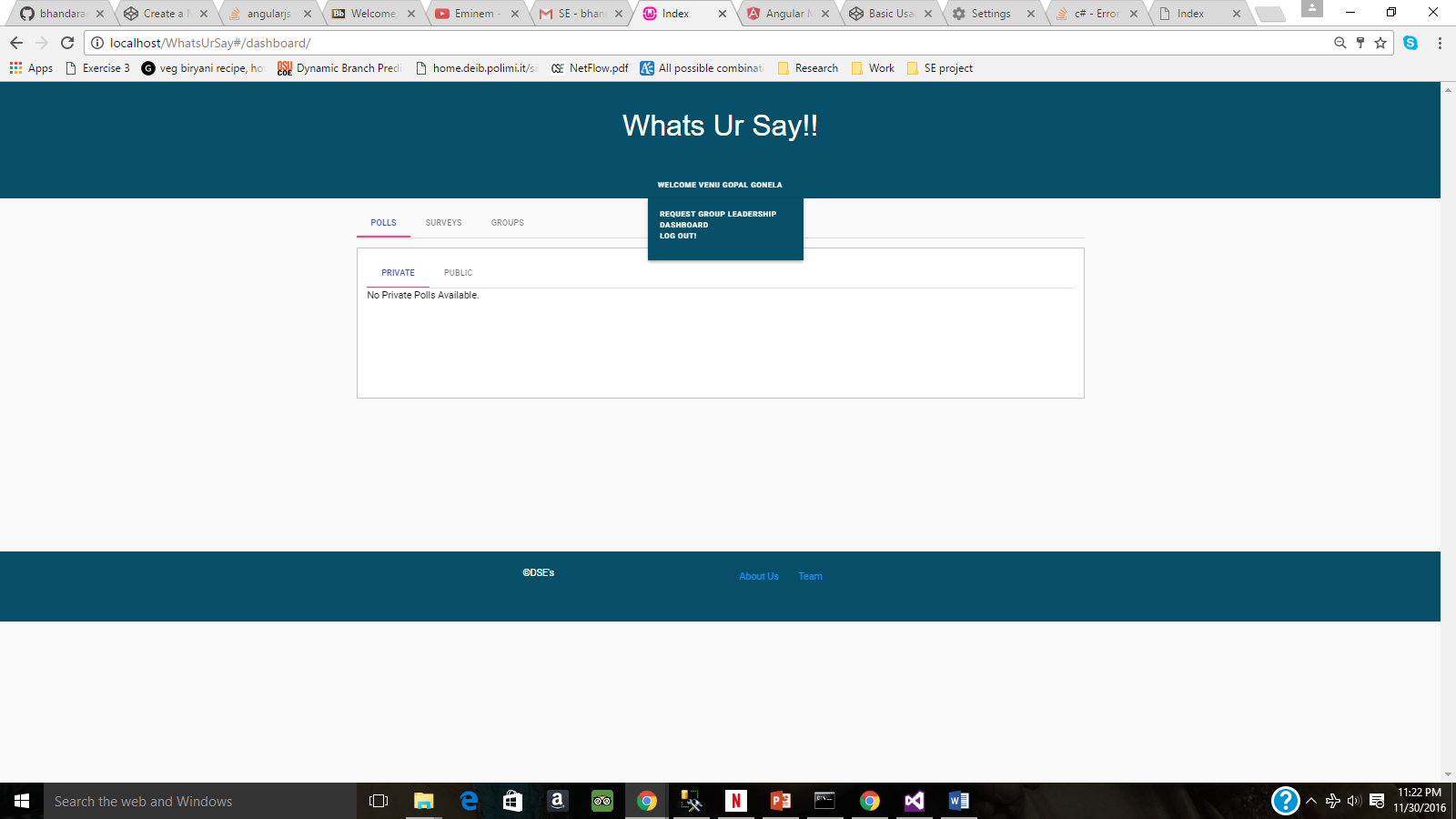


Figure 20: Request Group Leadership option page

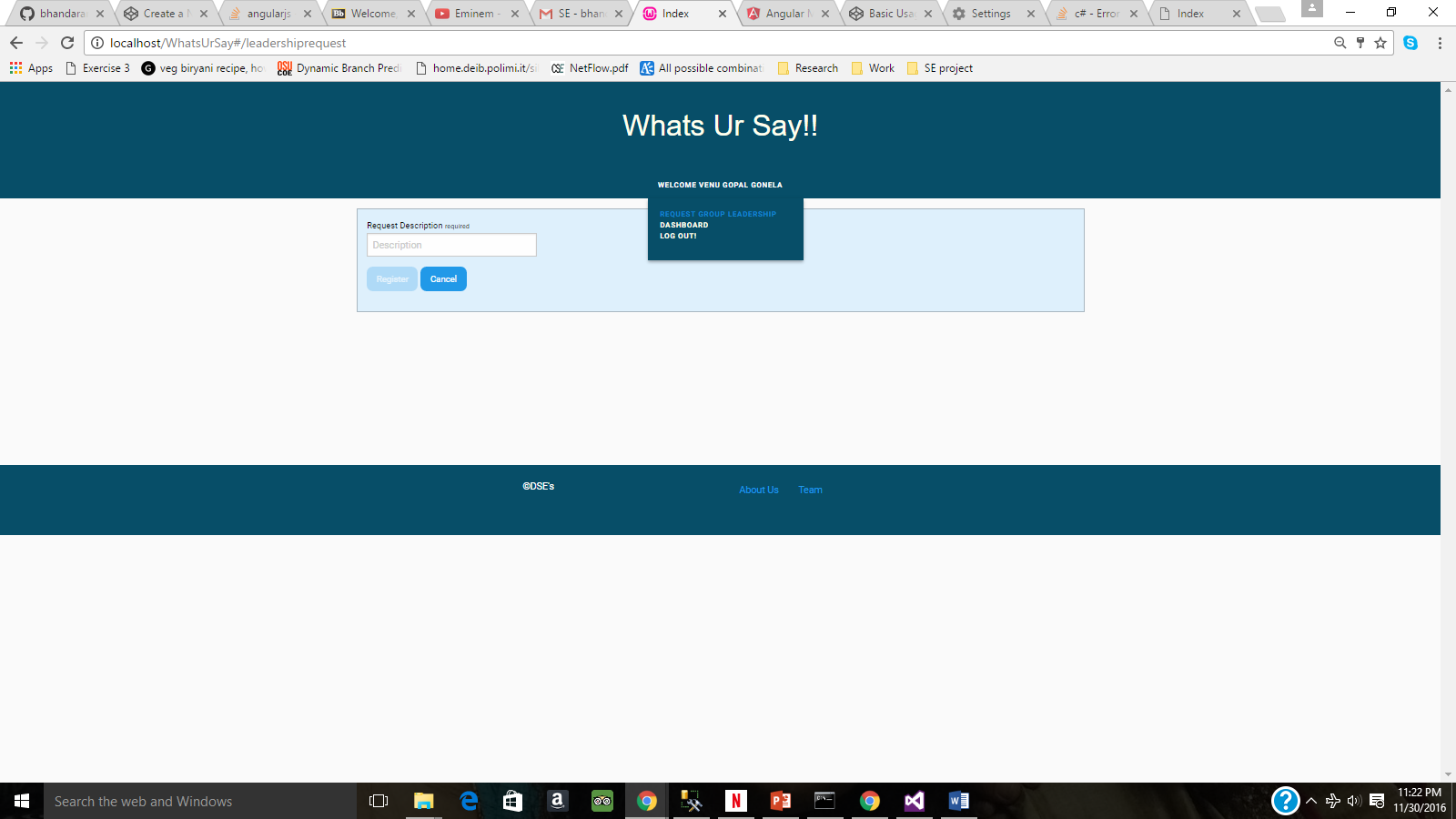


Figure 21: Request for group leadership page

**Logout:**

Clicking on ‘Logout’ will log out the user safely from the system

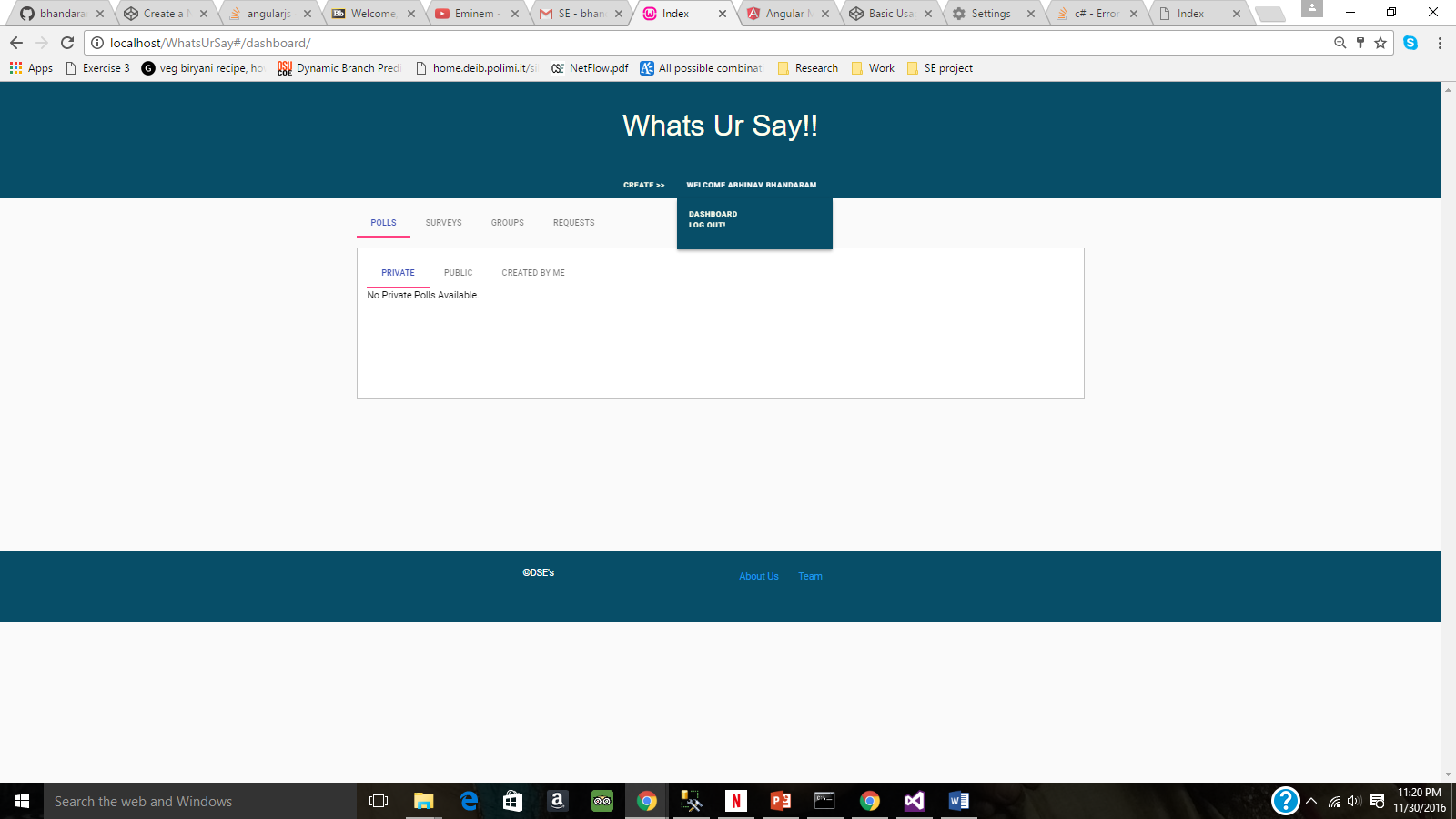


Figure 22: Page displaying an option to logout from the system

# Program compilation and execution instructions

1. Download the project ‘SEProject’ from Github
2. Before proceeding to compile the project, please execute the database script ‘WhatsUrSayTables.sql’ on SQL server, to create the database in the system. ‘WhatsUrSayTables.sql’ database script is present in the downloaded project in the path ‘SEProject\Source\Dev\DBScripts’
3. In the downloaded ‘SEProject’, navigate to the path ‘SEProject\Source\Main\WhatsUrSay’, where the project’s solution file ‘WhatsUrSay.sln’ is present
4. Open the solution ‘WhatsUrSay.sln’ using ‘Visual Studio’ in administrator mode.
5. Under the root directory of the project, a configuration file called ‘Web.config’ is present. The ‘connectionStrings’ portion of the Web.config file needs to be edited such that the connection string is pointing to your local database.

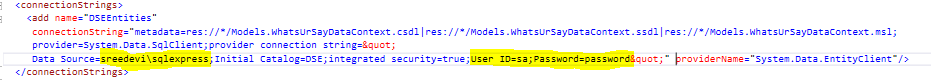


Figure: Connection string example

In the above figure, the ‘connectionString’ of ‘Web.config’ is edited such that the ‘Data Source’ attribute is pointing to my local server.

1. Now, on the menu bar of the visual studio, click on ‘Debug’ menu and then click on ‘Start Debugging’ option that compiles and executes the program
2. Once the program is executed, the new tab is opened on the browser and our web application launches.

Note: As the project’s execution is dependent on configuration settings and IIS server settings, you may face environmental issues in setting up the project and executing the same. Please let us know if you find any environmental issues with the project compilation and execution.

# Test suite compilation and execution instructions

1) Download the “WhatsUrSay.Test” project along with the main project.

2) make sure you refer the project ‘WhatsUrSay’ in “WhatsUrSay.Test” project references. And add all the other references that are need for testing. (As we have done the Microsoft Unit Testing for our project: import all the Microsoft Unit Testing Famework references)

3) Before proceeding to compilation of the project, make sure you have executed the database Script ‘WhatsUrSayTables.sql’ and there are values in all the database tables.

4) Build the “WhatsUrSay” main project.

5) If there are any issues debug by clicking Ctrl+R, Ctrl+T.

6) In the tool bar->Test->click on run AllTests

7) The results will be shown in the Test Explorer.

# Automated Test Cases

Below are the details of automated test cases that are written using Microsoft Unit Testing framework. These test cases are used in testing the Web API controller methods

## TestAnswerController.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using WhatsUrSay.Controllers;

using WhatsUrSay.DTO;

using WhatsUrSay.Models;

namespace WhatsUrSay.Tests

{

[TestClass]

public class TestAnswerController

{

[TestMethod]

public void GetAllAnswers\_ShouldReturnAllAnswers()

{

var testAnswer= GetTestAnswer();

var controller = new AnswerController();

var result = controller.GetAnswers().ToList<Answer>();

Assert.IsNotNull(result);

}

private List<Answer> GetTestAnswer()

{

var testAnswer = new List<Answer>();

testAnswer.Add(new Answer { id = 7, description= "yes", question\_id=3, count=0 });

return testAnswer;

}

}

}

## TestGroupController.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using WhatsUrSay.Controllers;

using WhatsUrSay.Models;

namespace WhatsUrSay.Tests

{

[TestClass]

public class TestGroupController

{

[TestMethod]

public void GetAllGroups\_ShouldReturnAllGroups()

{

var testGroup = GetTestGroup();

var controller = new GroupsController(testGroup);

var result = controller.GetGroups() as DbQuery<Group>;

Assert.AreEqual(testGroup.Count, result.Count());

}

private List<Group> GetTestGroup()

{

var testGroup = new List<Group>();

testGroup.Add(new Group { id = 1, name="se team",createdby=1 });

return testGroup;

}

}

}

## TestLoginController.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using WhatsUrSay.Controllers;

using WhatsUrSay.Models;

namespace WhatsUrSay.Tests

{

[TestClass]

public class TestLoginController

{

[TestMethod]

public void TestLogin\_Controller()

{

var testLogin = GetTestLogin();

var controller = new LoginController(testLogin);

UserInput user = new UserInput();

user.uName = testLogin[0].name;

user.uPassword = testLogin[0].pwd;

var result = controller.Login(user);

Assert.IsNull(result);

}

private List<User> GetTestLogin()

{

var testUser = new List<User>();

testUser.Add(new User { id = 1, name = "raj", emailId = "raj@gmail.com", pwd = "Raj1", status = "active" });

return testUser;

}

}

}

## TestPollController.cs

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System;

using System.Collections.Generic;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using WhatsUrSay.Controllers;

using WhatsUrSay.Models;

using System.Web.Http.Results;

using System.Data.Entity.Infrastructure;

using WhatsUrSay.DTO;

namespace WhatsUrSay.Tests

{

[TestClass]

public class TestPollController

{

[TestMethod]

public void GetAllPoll\_ShouldReturnAllPoll()

{

var testPoll = GetTestPoll();

var controller = new PollController(testPoll);

var result = controller.GetAllPolls() as DbQuery<ActivityDTO>;

Assert.AreEqual(testPoll.Count, result.Count());

}

[TestMethod]

public void GetPoll\_ShouldReturnCorrectPoll()

{

var controller = new PollController(GetTestPoll());

var result = controller.GetPoll(2);

Assert.IsInstanceOfType(result, typeof(DbQuery<ActivityDTO>));

}

private List<Activity> GetTestPoll()

{

var testActivity = new List<Activity>();

testActivity.Add(new Activity { id = 2, heading = "ExamPattern", description = "FinalExamPatternForSe", type = 1, category = 1, createdby = 1, results\_published = null });

testActivity.Add(new Activity { id = 1, heading = "DoYouWantClass", description = "IsSoftwareEngineeringClassRequiredon20th", type = 1, category = 1, createdby = 1, results\_published = null });

return testActivity;

}

}

}

## TestQuestionController.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using WhatsUrSay.Controllers;

using WhatsUrSay.Models;

namespace WhatsUrSay.Tests

{

class TestQuestionController

{

public void PostQuestiControlleron\_ShouldPostQuestions()

{

var testQuestion = GetTestQuestions();

var controller = new QuestionController(testQuestion);

var result = controller.PostQuestion() as List<Question>;

Assert.AreEqual(testQuestion.Count, result.Count);

}

private List<Question> GetTestQuestions()

{

var testActivity = new List<Question>();

testActivity.Add(new Question { id = 1, description = "ClassNeeded", activity\_id=1});

return testActivity;

}

}

}

## TestSurveyController.cs

using System;

using System.Collections.Generic;

using System.Threading.Tasks;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System.Linq;

using System.Text;

using WhatsUrSay.Controllers;

using WhatsUrSay.Models;

using System.Web.Http.Results;

namespace WhatsUrSay.Tests

{

[TestClass]

public class TestSurveyController

{

[TestMethod]

public void GetAllsurveys\_ShouldReturnAllsurveys()

{

var testSurvey = GetTestSurvey();

var controller = new SurveyController(testSurvey);

var result = controller.GetAllSurveys() as List<Activity>;

Assert.AreEqual(testSurvey.Count, result.Count);

}

[TestMethod]

public void GetSurvey\_ShouldReturnCorrectSurvey()

{

var testSurvey = GetTestSurvey();

var controller = new SurveyController(testSurvey);

var result = controller.GetSurvey(3);

Assert.IsNotNull(result);

Assert.AreEqual(testSurvey[0].heading, result.heading);

}

[TestMethod]

public void Getsurvey\_ShouldNotFindsurvey()

{

var controller = new SurveyController(GetTestSurvey());

var result = controller.GetSurvey(3);

Assert.IsInstanceOfType(result, typeof(Activity));

}

private List<Activity> GetTestSurvey()

{

var testActivity = new List<Activity>();

testActivity.Add(new Activity { id = 3, heading = "Professor Feedback", description = "SE Professor feedback", type = 1, category = 2, createdby = 1, results\_published = null });

return testActivity;

}

}

}

## TestUserController.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Web.Http.Results;

using WhatsUrSay.Controllers;

using WhatsUrSay.Models;

using WhatsUrSay.Repositories;

namespace WhatsUrSay.Tests

{

[TestClass]

public class TestUserController

{

[TestMethod]

public void GetAllUsers\_ShouldReturnAllUsers()

{

var testUser = GetTestUser();

var controller = new UserController(testUser);

var result = controller.GetUsers().ToList<User>();

Assert.IsNotNull(result);

}

private List<User> GetTestUser()

{

var testUser = new List<User>();

testUser.Add ( new User { id = 1, name = "raj", emailId = "raj@gmail.com", pwd = "Raj1", status = "active" } );

return testUser;

}

}

# Testing Experiences

* The time allocated was not sufficient to write unit test cases for all implemented methods
* Faced challenges while writing the Microsoft Unit Testing TestMethods for the controller methods which interact with the application’s database.
* Challenges:
  + New to the test framework
  + Identifying the test scenarios
  + Debugging the root cause in case of a test failure that is supposed to pass

# Implemented features

Below are the details of the features that are implemented successfully in our current version of the project.

## Create Poll:

* + This feature enables the creation of a Poll by authorized users (Admin and Group Leaders).
  + The types of polls supported by the system: Private and Public.
* Private Polls: The polls in which only specific groups of people can participate in.
* Public Polls: The polls in which any user of our system can participate in.

## Create Survey:

* + This feature enables the creation of a Survey by authorized users (Admins and Group Leaders).
  + The types of polls supported by the system: Private and Public.
* Private Surveys: The surveys in which only specific group of people can participate in.
* Public Surveys: The surveys in which any user of our system can participate in.

## Create User Group:

* This feature enables the creation of a User Group by authorized users (Admins and Group Leaders).
* The User Groups are created to categorize the people. This ensures a particular category of people to participate only in their relevant private polls/surveys.

## Add User to User Group:

* This feature enables the addition of new users to a particular user group by authorized users (Admins and Group Leader that has created the group).
* Once a User Group is created, the authorized users can add as many users as needed to that group

## Remove User from User Group:

* This feature enables the removal of a user from a user group by an authorized user.

## Participate in Poll:

* This feature is used by a user to participate in a particular poll by submitting his response to the poll question.

## Participate in Survey:

* This feature is used by a user to participate in a particular survey by submitting his response.

## Request for Group Leadership:

* This system feature is used by the normal users to attain the role of a ‘Group Leader’

## Respond to Group Leadership Request:

* This system feature enables the Admin to either approve or reject the Group Leadership request from a normal user.
* The admin makes a decision based upon the description or reason behind the user’s request.

## View Poll Results:

* This feature is used by a user to view the results of a particular poll that he is authorized to or has participated in.

## View Survey Results:

* This feature is used by a user to view the results of a particular survey that he/she is authorized to or has participated in.

## Publish Results:

* This feature is used by authorized users to publish the results of a particular poll/survey

## Register:

* This feature is used by a person to register in our system

## Sign In:

* This feature enables an authorized person to sign in to the system.

## Sign Out:

* This feature is used by a person to safely exit from the current user session of the application

# Limitations

There are few limitations in our project that correspond to the implementation of features and the way of implementing the same.

The limitations are imposed by the below specified constraints.

## Time Constraint:

* The property ‘Root Admin’ is not introduced in application. Hence only admin, group leader and normal user roles exist
* The feature ‘Create Admin’, which enables the root admin to create an additional admin
* The feature ‘Forgot Password’, which enables a person to reset his password in case if he forgets it.
  + Publishing results of polls/surveys through email is not supported
    - We are publishing the results on user dashboard

## Platform Dependent Constraint:

* + **Session Management**
    - Server side session management using Session is not supported by ASP.NET Web API. Therefore we had to come up with an approach on client side to implement session using Angular Session Storage

# Future Enhancements

We plan to further enhance our project by including the below specified features.

1. Incorporate the user role ‘Root Admin’ in the system. The current user roles supported are only ‘admin’, ‘group leader’, ‘normal user’.

The ‘Root Admin’ will be given the privilege to create any number of admins in the system when required, along with all the privileges of an ‘admin’.

1. ‘Create Admin’ feature: This feature will provide an option to the root admin to create an additional admin in the system.
2. Currently the results are published only on the user dashboard. It will be convenient and time-saving to the users to know the poll/survey results in an email instead on login into the system just to view the results. Hence, we plan to implement the feature of publishing the results of a poll/survey via email.
3. Feature ‘Forgot Password’: This feature is useful to a person to reset his password in case if he/she forgets the password.

# Minutes of Meeting

1. **Meeting on:** 10/14/2016

**Location:** Library

**Time:** 8:15 PM – 10:00 PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Discussed about the project scope.

2. Discussed about the modules to be implemented or the project.

1. **Meeting on:** 10/18/2016

**Location:** In class 157

**Time:** 10:50 AM – 11:20 AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Created the Database for the project.

1. **Meeting on: 1**0/20/2016

**Location:** In class 157

**Time:** 10:50 AM – 11:20 AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Divided the modules to be implemented within the group.

1. **Meeting on:** 10/25/2016

**Location:** In class 157

**Time:** 10:50 AM – 11:20 AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Started implementing the controller required for the project.
2. **Meeting on:** 10/27/2016

**Location:** In class 157

**Time:** 10:50 AM – 11:20 AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Discussed about the design for the project.

6) **Meeting on:** 10/30/2016

**Location:** Willislibrary

**Time:** 10 AM – 12:45PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Implemented the UI for the project: The login.html, Home.html.

**7) Meeting on:** 11/01/2016

**Location:** In willi

**Time:** 10:50 AM – 11:20 AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Worked on the ViewSurveyResults.html.
2. Worked on the Database Scripts.

**8) Meeting on:** 11/02/2016

**Location:** Willis Library

**Time:** 6:30 PM – 10:15 PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Worked on the Register.html page
2. Worked on the UserAnswerController.cs

**9) Meeting on:** 11/03/2016

**Location: In class**

**Time:** 11:00 AM – 11:20AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Worked on Login Controller
2. Worked on the Home Controller.

**10) Meeting on:** 11/08/2016

**Location: In class**

**Time:** 11:00 AM – 11:20AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Resolved few issues that came while integrating the project .

**11) Meeting on:** 11/10/2016

**Location: In Willis Library**

**Time:** 5:30PM – 9 PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Created the UserController and Group controller.

**12) Meeting on:** 11/15/2016

**Location: In class**

**Time:** 11:00 AM – 11:20AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1) Discussed about how to design the UI for the project.

**13) Meeting on:** 11/17/2016

**Location: In Libray**

**Time:** 5:20 PM – 7:20PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Created the poll controller functionality.

2. Created the Poll.html page.

**14) Meeting on:** 11/17/2016

**Location: In Library**

**Time:** 4:00 PM – 7:20PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Designed the Survey Controller.

2. Designed the Survey.html page.

**15) Meeting on:** 11/22/2016

**Location: In Library**

**Time:** 10 AM – 12:20PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Implemented the Error.html file

2. Implemented the dashboard functionality.

**16) Meeting on:** 11/24/2016

**Location: In Willis Library**

**Time:** 11:00 AM – 11:20AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Worked on setting up the Microsoft Unit Testing Framework.

2. Worked on participate poll functionality.

**17) Meeting on:** 11/25/2016

**Location: In Willis Library**

**Time:** 5:10 PM – 8:40PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Worked on the database issues.

2. Worked on Participate survey functionality.

**18) Meeting on:** 11/26/2016

**Location: In class**

**Time:** 5:15 PM – 2:15AM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Prepared slides for the final group presentation.

2. Worked on the Edit Group functionality.

**19) Meeting on:** 11/27/2016

**Location: In Willis Library**

**Time:** 5:00PM – 10:15PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Worked on the integration of the project onto Github.

2. Discussed about the final presentation.

**20) Meeting on:** 11/28/2016

**Location: In Willis Library**

**Time:** 4:30 PM – 10:20PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

**1.** Worked on Microsoft Unit Testing for the project on the files TestGroupController,TestUserController.

**21) Meeting on:** 11/29/2016

**Location: In Willis Library**

**Time:** 5:30PM – 11:30PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sreedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Discussed about the deliverable III report.

2. Worked on unit testing on the files TestPollController, TestSurveyController, TestLoginController.

**22) Meeting on:** 11/30/2016

**Location: In Willis Library**

**Time:** 5:15 PM – 12:00PM

**Team Members/ Attendees**:

Abhinav Bhandaram (Team LEAD)

Rajashekar Goud Korakoppula

Sr eedevi Koppula

Kaza Nikhitha

**Topics discussed:**

1. Worked on the integration of the project in Github.

2. Worked on deliverable-III report.

# Member contribution table

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution (%) | Note |
| Sreedevi | Components Written:   1. UI:    1. CreateGroup.html    2. CreatePoll.html    3. ParticipateInPoll.html    4. ViewPollResults.html 2. AngularJS controllers: 3. GroupController.js 4. PollController.js 5. Web API controllers: 6. ActivityDTOController.cs 7. ActivityGroupDetailsController.cs 8. AnswerConroller.cs 9. PollAnswerDetailsController.cs 10. PollController.cs 11. Other: 12. Assisted in Database creation   Report Contribution:   1. Implemented Features 2. Limitations 3. Contribution Table | 25 | NA |
| Nikhitha | Components Written:   1. Web API controllers: 2. SurveyController.cs 3. QuestionController.cs 4. Database Script 5. Unit Test Cases 6. TestAnswerController.cs 7. TestGroupController.cs 8. TestLoginController.cs 9. TestPollController.cs 10. TestQuestionController.cs 11. TestSurveyController.cs 12. TestUserController.cs   Report Contribution:   1. Minutes of Meeting 2. Automates Test Cases 3. Testing Experiences | 25 | NA |
| Rajashekar | Components Written:   1. UI: 2. EditUserGroup.html 3. CreateSurvey.html 4. ParticipateInSurvey.html 5. ViewSurveyResults.html 6. AngularJS controllers: 7. GroupController.js 8. SurveyController.js 9. ParticipateSurveyController.js 10. Web API controllers: 11. GroupsController.cs 12. UserAnswerController.cs 13. Other: 14. Integrated ‘Edit User groups’ module   Report Contribution:   1. User Manual | 25 | NA |
| Abhinav | Components Written:   1. UI: 2. Error.html 3. Dashboard.html 4. Home.html 5. Login.html 6. Register.html 7. AngularJS controllers: 8. HomeController.js 9. LoginController.js 10. RegisterController.js 11. Web API controllers: 12. HomeController.cs 13. LoginController.cs 14. UserController.cs 15. Other: 16. Components Integration 17. Project Configuration settings 18. Session Management   Report Contribution:   1. Program compilation and execution instructions | 25 | NA |