# Individual Assignment Specifications

# Team Bharat

**Project:** iHunt
**Customer:** Sayma Akther

| | | |
|---|---|---|
| **Project Coordinator** | : | Megha Vashisht |
| **Quality Assurance Czar** | : | Chandra Sekhar Challapilla |
| **Video-Demo Creator(s)** | : | Megha Vashisht, Jobin Sunny |
| **Demo-Booth Operator** | : | Sree Lasya Vallabhaneni |

## Individual Tasks:

### Developer: Chandra Sekhar Challapilla

**Task 1: Registration**
**Task Description:** A user should be able to register to the website. On arriving to the homepage, the user should see a “Register” button that takes the user to the Registration page. The Registration page will have fields asking for email address and password (and a confirm password field). The page will also ask the user if he/she is a student or a faculty member and will register the user to the database. On registering to the app, the user should be redirected to the “My Profile” page so that he/she can start creating his/her profile on the app.

**How to Evaluate:** There will be test cases that run the code and test for correctness. On clicking on the “Register” button, the user should be directed to the Register page and that page should have the fields mentioned above. The database should also reflect the registered users.

**Task 2: Login**
**Task Description:** A user should be able to login to the website by finding a “Login” button on the homepage. On clicking on the login button, the user should be able to login using his/her credentials. The login page should also have a “Forgot your password” link that will allow users to retrieve their password. On logging in, the user should be directed to his dashboard page.

**How to Evaluate:** There will be test cases showing the users logging in. On logging in using the web app, the user should be directed to the correct page and should be logged in. Being logged in can be proved by the name on the user's dashboard.

**Task 3: Logout**

**Task Description:** A user should be able to click on the "Logout" button and logout of the application at any time. On clicking the logout button, the user's session information should be cleared and the user should be safely logged out.

**How to Evaluate:** There will be test cases testing the functionality of the logout. The website should also log out and show a visual cue when the user has successfully logged out.

**Task 4: Dashboard page**

**Task Description:** On successfully logging into the application, the user should be redirected to a dashboard page. The dashboard page will have the name of the user on the top right, will have the user's scores and contributions listed. The dashboard page will also have links to "My Profile", "My Projects", "Search Projects" and other relevant links.

**How to Evaluate:** There will be test cases testing the functionality of the dashboard links. The page should also correctly display the name and projects assigned to that particular user. There should be empty links on the dashboard page. These links will later (in Iteration 2) show you the details within the dashboard itself.

**Task 5: Forgot your password**

**Task Description:** The user can click on the "Forgot your password" link if he/she forgets his/her password and then retrieve the password.

**How to Evaluate:** Unit tests will be able to test some of the functionality. The rest of the functionality can be tested by the website by trying to change a user's password securely and correctly.

**Task 6: Model Classes**

**Task Description:** Set up all the model classes for users and authentication.

**How to Evaluate:** Test cases to check if the model classes have basic CRUD functionality working.

# Developer: Megha Vashisht

**Task 1: Create Project**

**Task Description:** A user should be able to create a new project. On creating a new project, the user will fill out a form with all the necessary details and on clicking the “Create Project” button, will have successfully created a new project. The details for the form are the attributes from the model diagram.

**How to Evaluate:** The website should have a button on the dashboard for the user to create a new project. On clicking the button, the form should be shown and after submitting the form, the new project should be in the database. I will also be making test cases for the forms to make sure everything is working fine.

**Task 2: Update Project**

**Task Description:** A user should be able to update his or her projects. On looking at the project page, the user can click the “Update Project”. Only the founding member for that project can see and has access to Update the details of the project. The form generated from updating project should be auto-filled with the current details of the project. On updating, the changes should be in the database as well.

**How to Evaluate:** The website should have a button that lets the user go to the Update project form. On submitting the form, the new details should be in the database as well. This can be checked either by viewing the project again, or inspecting the database values. I will also have test cases to test the edit form.

**Task 3: View Project**

**Task Description:** A user will be able to view any of his or her projects. After clicking on a particular project, the user should be able to see all the details of that project. The details are the attributes in the UML diagram for the Project. The view page should have all the details listed in a readable format and should have some sensible default values for fields that are empty.

**How to Evaluate:** The website will generate a page for a particular project and have all the details for that project being displayed. The details should be the correct ones and should print default values to replace empty fields. I will also have test cases to test that the database models are working correctly.

**Task 4: Delete Project**

**Task Description:** A user can delete a particular project only if he/she is the founding member of that project. The user should find a delete project button when viewing a project and be able to delete all the data related to that project. One complexity when deleting the project details is to delete all the team, tasks and other details for that project.

**How to Evaluate:** The web app should have a delete button which should delete the project from the database. It should also delete all other data associated to that project from other tables in the database. After deleting a project, that project should not show up in the “My

Projects” list for the founding member or any other members. I will also provide test cases to automate some of the process and test it.

**Task 5: Model Classes**

**Task Description:** Set up all the model classes for project and related models.

**How to Evaluate:** Test cases to check if the model classes have basic CRUD functionality working.

# Developer: Jobin Sunny

**Task 1: View Profile**

**Task Description:** As a user, I should be able to see my own profile and the profile of other users in the website. On my dashboard, there will be a link to view my profile, and on clicking a link to my profile, I should be redirected to a page where I can see all my profile details. I should also be able to click on other users' names and get to their particular profile. Only faculty members can see all the details of students whereas students can only see a basic profile of others.

**How to Evaluate:** Apart from test cases that will test and run the code for viewing profiles, the website can also show the profile details. On seeing a profile, one can check if the details on that profile is correct. Similarly, on looking at other people's profiles, one can check if the profiles show only the right amount of information as specified above.

**Task 2: New Profile**

**Task Description:** After registering to the website, the user should be automatically redirected to a new profile page. This page contains all the details to be filled to complete a profile. On submitting the form, the information from the form should be added to that profile details and saved in the database.

**How to Evaluate:** Apart from the test cases that will test the code, one can check the website to make sure everything is working correctly. A new user should be redirected to the New Profile page and should see a form with all the details for a profile. On submitting the form, the user can check his/her profile page to find all the details added to his/her profile. Additionally, by looking at the database itself, one can check to see if the details were added properly.

**Task 3: Update Profile**

**Task Description:** A user can update or edit his/her profile at any point. The user can click on an "Update Profile" link and see a form with default values of the saved, current details. On pressing "Update", the form should submit and save all the new values that were in the form.

**How to Evaluate:** Apart from the test cases that will test the functionality, the website should also show you an updated version of the profile when viewing after changing details. On submitting form, the details should be saved on the database and can also be checked with some outside tools like the SQLite3 Manager Firefox extension.

**Task 4: Search Projects**

**Task Description:** A user can search all in-progress projects to find new projects to join. On clicking the search projects link, the user should go to the "Search Projects" page which can show all the projects that are currently being worked on. One thing to note is that, this page should not be showing projects that are completed even though completed projects will be in the database. The user can search by research category or by name to narrow down search results to interesting projects.

**How to Evaluate:** Apart from the test cases that will test functionality, the website should show all current projects that are being worked on. The list of current projects can be looked at from the database and the website should show just those projects. On looking for a particular project, the project lists should show only the projects from a particular research topic or name.

**Task 5: Model Classes**

**Task Description:** Set up all the model classes for Profile and related models.

**How to Evaluate:** Test cases to check if the model classes have basic CRUD functionality working.

# Developer: Sree Lasya Vallabhaneni

**Task 1: Add Team Member**

**Task Description:** A project can have many team members. On the profile page, if the user is the founding member of that project, he/she can add team members to the project. There will a dropdown of members that he/she can add to the project and can add them one by one.

**How to Evaluate:** There will be unit tests that test the code. We can also add team members on the website itself. By going to the project page, there should be a “Add Team Member” button (only the founding member can see this button) and on clicking that, there will be a dropdown of all potential members. The founding member can add anyone and the new team member should be visible on the project page under the team listing.

**Task 2: Remove Team Member**

**Task Description:** A founding member can always delete a team member from the project. There will be a delete button next to a team member’s name. On clicking that, that team member will be deleted from the database. The team member will not receive any points or any deductions as the founding member is removing the team member.

**How to Evaluate:** There will be unit tests that test the code. We can also check if the delete button from the website works correctly. After deleting a member from a team, he/she should not be in the list of team members. The database should also show that the particular team member is not part of the team anymore.

**Task 3: Complete Project**

**Task Description:** The founding member of a project can mark a project as complete. After marking a project as complete, the founding member can give points and money to the team members who contributed. These points and money allow team members to build reputation for future projects.

**How to Evaluate:** There will be unit tests that test the code. We can also check via the website if on clicking the “Complete Project” button, whether the site goes to the “Award Contribution” page and if the project is marked complete in the database.

**Task 4: Award Contribution**

**Task Description:** After completing a project, the founding member can award points and money to the team members. The founding member will be redirected to a screen with each team member and a field to enter how much points and money he wants to award each member. After awarding points and money, each member’s scores should be changed accordingly.

**How to Evaluate:** There will be unit tests that test the code. We can also see if the website redirects properly from the complete project to the award contribution page. We can check if,

after awarding points to the team, the team members have their scores updated with the new points and money they just received. The database should also show the updated values and show that how much a member received from a particular project.

**Task 5: Model Classes**

**Task Description:** Set up all the model classes for team and other models.

**How to Evaluate:** Test cases to check if the model classes have basic CRUD functionality working.