Iteration 2 - Individual Assignment Specifications

Team Bharat

Project: iHunt
Customer: Sayma Akther

Project Coordinator	:	Megha Vashisht
Quality Assurance Czar	:	Sree Lasya Vallabhaneni
Video-Demo Creator(s)	:	Sree Lasya Vallabhaneni
Demo-Booth Operator	:	Megha Vashisht

Individual Planned Tasks:

Developer: Chandra Sekhar Challapilla

Task 1: Join Project Request
Task Description: A user should be able to make a request to join a project website. On arriving to a project view page, the user should see a “Join Project” that creates a request to the founding member of the project. User should not see this button if he is already a founding member or if he is already an existing team member. Once the request is approved by the founding member, the user becomes a team member and only then he will be able to only see the iterations of that project. As long as the request is made, the button will be disabled.

How to Evaluate: On clicking on the “Join Project” button, a request will be made to the founding member user. The database should also reflect the user’s requests.

Outcome of Task: There will be test cases testing the functionality of the requests. On using the website, one can manually send join request to a project that he is not on. On sending a join request, one can see the join button will be disabled and text changes to join request sent. User cannot see any information until the request is approved. The database also reflects the change to show the addition of the join request.

Task 2: Leave Project Request
Task Description: A team member should see a leave project button on a project he is working on. On clicking on the leave project button, the user’s request is sent to the founding member. User should not see this button if he is a founding member or if he is not an existing team member. Once the request is approved by the founding member, the user will be no longer a team member and he will see a join project button.

How to Evaluate: On clicking on the “Leave Project” button, a request will be made to the founding member user. The database should also reflect the user’s requests.

Outcome of Task: There will be test cases testing the functionality of the requests. On using the website, team member can manually send a leave request to a project that he is not on. On sending a leave request, one can see the leave project button will be disabled and text changes to leave request sent. User cannot see any information until the request is approved. The database also reflects the change to show the addition of the leave request.

Task 3: My requests

Task Description: A founding member should be able to see all the request he received for his projects in ‘My Requests’ page. Requests can be approved or rejected by him. He should be also able to see requests previously approved or rejected.

How to Evaluate: There will be test cases testing the functionality. Clicking ‘My Requests’ link in the sidebar should display the above functionality. The database should also reflect the user’s requests.

Outcome of Task: There will be test cases testing the functionality of the requests. On using the website, team member can click my requests to see all the requests he has sent along with their statuses. A founding member can additionally approve or reject different type of requests he has received for his projects. Once a request is approved or rejected by him, he cannot change it. If a join request is approved, the user who sent the request becomes the team member for the requested project. If a leave is approved by the founding member, the user who sent the request will be no longer be a team member. The database also reflects the change to show all current user’s requests or founding member requests.

Task 4: Revise Summary for Tasks and Requests on Dashboard page

Task Description: Dashboard page should show the summary of Tasks and Requests by their respective categories.

How to Evaluate: There will be test cases testing the functionality of the dashboard links. The page should also correctly display the tasks and requests assigned to that particular user. The database should also reflect the user’s requests.

Outcome:

- By visiting the home page at http://localhost:3000 and clicking the login link (located at the top right in the navigation), you can login with seeded user details (username: cchllpll@memphis.edu, password: topsecret) to the site as student. On successfully logging in, you will be redirected to dashboard page with a green coloured notice that says “Signed in successfully.”.
- Since seeds are in place, The dashboard should show the dynamic summary counts of ‘My projects’ , ‘My Tasks’, ‘My Requests’ ’and ‘My Contributions’ widgets correctly.

Developer: Megha Vashisht

Task 1: Project Research Fields

Task Description: A founding member should be able to add and remove a research field for a project. Current research fields should be displayed for a project. Edit option should allow user edit the existing research fields.

How to Evaluate: The project page should show the current research fields and edit research fields button should take you to a new page to add/remove research fields.
I will also be making test cases for the forms to make sure everything is working fine. The database should also reflect the project's research fields.

Outcome:

- After logging in with seeded user details (username: mvshisht@memphis.edu, password: topsecret), click on 'My projects' in the left sidebar.
- Then select the 'Edit' option for the project Authentication Framework for which Megha is the founding member.
- Once selected it take us to a edit project form. Select the 'Add' option for the Research fields.
- Now you can add or delete research fields from a list of research fields.
- When back is selected and we now choose the 'View' project option for the project we can now see all the research fields we just selected, reflected in the database.

Task 2: Complete Project Workflow

Task Description: A founding member should be able to mark a project as complete after adding their contributions. The completed project will no longer have access to modify, delete tasks and iterations.

How to Evaluate: The website should have a button on Mark project as complete page to handle this functionality. The database should also reflect the completed projects.

Outcome of Task: There are test cases for the model validations of projects. Apart from that, on using the website, one can mark a project as complete. Users will no longer be able to make changes to iterations or tasks of the project once a project is marked as complete. Such projects are reflected in the database and are not hardcoded into the web application.

Task 3: Delete Project Workflow

Task Description: As a founding member, whenever a project is deleted. The associated tasks and iterations should be deleted from the respective tables.

How to Evaluate: A delete button should be displayed to the founding member to handle this functionality. The database should also reflect the removed projects.

Outcome of Task: On using the website, one can delete a project. Only founding member can delete the project. Once a project is deleted, all its associated tasks and iterations will be removed from the database. Such projects are reflected in the database and are not hardcoded into the web application.

Task 4: Project Overview
Task Description: A team member should view the current iterations on the project page. A founding member should also see options to edit and delete an iteration. Unauthorized users should not be able to see this overview.

How to Evaluate: The project page should show the overview as per their right privileges.
Outcome of Task: On using the website, one view the project overview. A founding member and team members can see everything. A founding member can edit or delete anything about the project. A team member only read iterations while can create, edit and delete tasks of the iteration for the project. Normal users cannot see iterations or tasks of the project. Such projects are reflected in the database and are not hardcoded into the web application.

Developer: Jobin Sunny

Task 1: View Iterations
Task Description: As a founding member, I should be able to see all iterations I created for a project in the website. Only founding member and team members can see iterations for a project. Unauthorized users should not allowed to see list of iterations of a project.

How to Evaluate: There will be test cases testing the functionality of the iterations. A founding member would see iterations list in the project page. The database should also reflect the iterations for a project.

Outcome of Task: There are test cases for the model validations of iterations. Apart from that, on using the website, one can view all the iterations correctly for a particular project. These iterations are reflected in the database and are not hardcoded into the web application.

Task 2: Add Iteration
Task Description: As a founding member, I should be able to create iterations for a project in the website. Only founding member can create iterations. Team members should be able to view the iterations only. Unauthorized users should not allowed to access iterations.

How to Evaluate: There will be test cases testing the functionality of the iterations. A founding member would see create iteration button for his projects. The database should also reflect the iterations for a project.

Outcome of Task: There are test cases for the model validations of iterations. Apart from that, on using the website, one can manually add iterations to a project. On adding an iteration, one can see the tasks associated to that iteration and can see that it is correctly shown for that particular project. The database also reflects the change to show the addition of the new iteration.

Task 3: Edit Iteration

Task Description: As a founding member, I should be able to edit iterations I created for a project in the website. Only founding member can edit iterations. Unauthorized users should not allowed to edit iterations.

How to Evaluate: There will be test cases testing the functionality of the iterations. A founding member would see edit iteration button in the iterations page. The database should also reflect the iterations for a project.

Outcome of Task: There are test cases for the model validations of iterations. Apart from that, on using the website, one can edit a particular iteration. After editing, the iteration will show with the new name. Although the iteration has an association, the association cannot be changed and only the name can be changed, this is as intended. After editing, the database will also show the new iteration name correctly.

Task 4: Delete Iteration

Task Description: A founding member can always delete an iteration from the project. There will be a delete button next to a iteration name. On clicking that, that iteration will be deleted from the database.

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 an iteration from a project, he/she should not be in the list of iteration. The database should also show that the iteration is not part of the iterations anymore.

Outcome of Task: On using the website, one can delete a particular iteration. Once an iteration is deleted, it will not be visible on the iteration list for a particular project and will not be in the database either. This can be verified by looking at the database directly.

Developer: Sree Lasya Vallabhaneni

Task 1: View Task

Task Description: As a founding member, I should be able to see all tasks I created for a project in the website. Only founding member and team members can see tasks for a project. Unauthorized users should not allowed to see list of iterations of a project.

How to Evaluate: There will be test cases testing the functionality of the tasks. A team member would see tasks list in the iteration page. The database should also reflect the tasks for an iteration.

Outcome of Task: There are test cases for the model validations of tasks. Apart from that, on using the website, one can view all the tasks correctly for a particular iteration. Only team member or founding can see the tasks of a project. Other users are redirected to 403 page. Tasks are reflected in the database and are not hardcoded into the web application.

Task 2: Add Task

Task Description: As a founding member, I should be able to add tasks for an iteration in the website. Only founding member or team members can create task. Unauthorized users should not allowed to access iterations.

How to Evaluate: There will be test cases testing the functionality of the tasks. A team member would see create task button for his projects. The database should also reflect the tasks for an iteration.

Outcome of Task:

There are test cases for the model validations of tasks. Apart from that, on using the website, one can create tasks correctly for a particular iteration. Only team member or founding can create the tasks of a project. Other users are redirected to 403 page. Tasks are reflected in the database and are not hardcoded into the web application.

Task 3: Edit Task

Task Description: As a founding member, I should be able to edit tasks I created for a project in the website. Only team member can edit task. Unauthorized users should not allowed to edit iterations.

How to Evaluate: There will be test cases testing the functionality of the iterations. A team member would see edit task button in the iterations page. The database should also reflect the tasks for an iteration.

Outcome of Task:

There are test cases for the model validations of tasks. Apart from that, on using the website, one can edit tasks correctly for a particular iteration. Only team member or founding can edit the tasks of a project. Other users are redirected to 403 page. Tasks are reflected in the database and are not hardcoded into the web application.

Task 4: Delete Task

Task Description: A team member can always delete a task from the project. There will be a delete button next to a task name. On clicking that, that task will be deleted from the database.

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 task from an interaction, he/she should not be in the list of tasks. The database should also show that the iteration is not part of the iterations anymore.

Outcome of Task:
On using the website, one can delete tasks correctly for a particular iteration if he owns the task or if he is the founding member. Other users are redirected to 403 page. Tasks are reflected in the database and are not hardcoded into the web application.