

# SPRINT 3 PLAN

## PROJECT: TEAM FORMATION

---

### PROJECT SUMMARY:

The primary objective of this application is to streamline the process of student team formation based on key attributes such as gender, ethnicity, and programming proficiency. The professor requires a platform that enables the creation and management of a customizable form, allowing the collection of student data on these attributes. The application will allow the professor to define the attributes to be included, set weightages to guide the team formation algorithm, and specify a deadline for submission. Additionally, the professor will be able to modify the form and its associated weightages, as well as adjust the submission deadline. Upon closure of the submission period, the system will automatically form student teams while ensuring that diversity, inclusion, and proficiency considerations are met.

Students, as key stakeholders, will have access to the form through a secure login, allowing them to submit and update their responses prior to the deadline. Once submissions are finalized, the application will generate teams of four students, with provisions for smaller teams if necessary. The team formation algorithm will prioritize diversity by promoting balanced gender representation, ensuring that when females are present in a team, the algorithm will strive to include at least two females. It will also aim to avoid isolating students from underrepresented ethnic groups, grouping at least two individuals from these backgrounds within the same team where possible. By automating and optimizing this process, the application effectively addresses the needs of both the professor and students, ensuring fair, inclusive, and well-balanced team formation.

---

### TEAM ROLES:

**PRODUCT OWNER:** Ramya Reddy Gotika ([ramyagotika@tamu.edu](mailto:ramyagotika@tamu.edu))

**SCRUM MASTER:** Rithika Sapparapu ([rithikasapparapu@tamu.edu](mailto:rithikasapparapu@tamu.edu))

**TEAM MEMBERS:** Hitha Magadi Vijayanand, Phani Jyothi Kurada, Raja Karthik Vobugari, Sahithi Duppati, Sai Jaideep Reddy Mure, Uday Kumar Reddy Mettukuru

---

## CLIENT MEETING DETAILS:

**Date:** 21 October 2024

**Time:** 4:15 PM

**Place:** ZACH 424

In the meeting with the client to review the latest updates and gather feedback, the product owner demonstrated the updated deployed website. The client expressed satisfaction with the improvements made and noted that everything looks in place.

### Discussion Points:

1. **Website Review:** The client liked the overall design and functionality of the updated website, appreciating the enhancements made since the last meeting.
2. **Response Visibility:** The client requested the ability to view the responses filled out by students. This feature will be important for monitoring and evaluating student data submissions.
3. **General Feedback:** Aside from the visibility request, the client mentioned that the current implementation meets her expectations and looks solid.
4. **Implementation Timeline:** The client expects the back-end logic to be implemented during this sprint, focusing on core functionality and data processing.

### Regular Weekly Meeting Schedule:

The product owner has confirmed the recurring weekly meeting every Monday at approximately 4:15 PM in room ZACH 424.

**Meeting Link:** <https://meet.google.com/rue-uyno-swr>

---

# USER STORIES

**Feature:** Populate teams by gender - part 2

As a professor,  
So that I can ensure gender representation (female and other non-male genders) in teams,  
I want to populate teams by gender.

**Scenario:** Professor tests that system ensures at least two individuals from gender groups (female and other non-male genders) in teams

**Given** the submission period has closed

**When** the system processes student data for team formation

**Then** each team should include at least two individuals from gender groups (female and other non-male genders) if applicable

**Feature:** View Responses for a form

As a professor,  
I want to view student responses for any form.

**Scenario:** Professor views detailed student response

**Given** I am logged in as a professor

**And** I have already published a form

**And** students have submitted their responses

**When** I click on "View Responses"

**Then** I should see a list of student responses

**When** I select a response and click on "View Details"

**Then** I should see the detailed response

**Feature:** Populate teams by ethnicity - part 2

As a professor,  
So that I can ensure inclusivity in teams,  
I want the system to include at least two individuals from underrepresented groups.

**Scenario:** System includes underrepresented groups in teams

**Given** the submission period has closed

**When** the system processes student data for team formation

**Then** each team should include at least two individuals from underrepresented ethnic groups if applicable

**Feature:** Update Existing Form Responses On Editing The Form

**Scenario:** Form structure is modified after responses are received

**Given** I am editing the form structure (e.g., adding, removing, or modifying fields)

**When** I save the changes to the form

**Then** all existing responses should be updated to reflect the new form structure

**And** any new mandatory fields should have a default or blank value in existing responses

**And** any removed fields should no longer appear in existing responses

**Scenario:** Student's previous response is missing a newly required field

**Given** I have modified the form to add a new required field

**And** there are existing responses from students

**When** I review a response with the missing required field

**Then** I should see a prompt to fill in this field before marking the response as complete

**Scenario:** New validation rules are applied to existing responses

**Given** I am updating validation rules for certain fields in the form

**And** there are existing responses with data that might not meet the new validation criteria

**When** I apply the new validation rules to the form

**Then** I should be notified of any responses that do not meet the updated criteria

**And** I should have the option to edit or notify students to update their responses

**Scenario:** Form changes are saved without impacting response integrity

**Given** I am editing form fields that do not change the form's structure (e.g., changing field labels, making minor text edits)

**When** I save the form updates

**Then** the responses should not be impacted

**And** the updated field labels **and** text should display in all responses seamlessly

**Scenario:** Student edits their response after a form is updated

**Given** a student has already submitted a response to a form

**And** I make updates to the form structure or validation

**When** the student attempts to edit their response

**Then** they should see the updated form with new fields or validations

**And** they must complete any new required fields or comply with updated validations before resubmitting

**Feature:** Block Submissions After Deadline

As a professor,

So that I can ensure fairness,

I want to block students from submitting the form after the deadline has passed.

**Scenario:** Submissions are blocked after the deadline

**Given** I have set a deadline for the form

**And** the deadline has passed

**When** a student attempts to submit the form

**Then** they should see a message indicating the deadline has passed

**And** they should not be able to submit their responses

**Feature:** Team Shuffling with Diversity Constraints

**Background:**

**Given** the teacher has uploaded students with a variety of diversity attributes

**And** the teacher clicks on the "Generate Teams" button

**And** initial teams of 3 or 4 students have been created that meet diversity constraints

**Scenario:** Shuffling teams while maintaining diversity constraints with variable team sizes

**Given** teams of 3 or 4 students have already been created that meet diversity constraints

**When** the teacher clicks on the "Shuffle Teams" button

**Then** all teams should be reshuffled

**And** each team should still meet diversity constraints based on the defined factors

**And** no student should appear in multiple teams

**And** the teams should be unique from their previous arrangement

**Scenario:** Ensuring balanced representation by diversity attributes after shuffling

**Given** initial teams of 3 or 4 students that meet diversity constraints have been generated

**When** the teacher shuffles the teams

**Then** each team should contain a balanced mix of diversity attributes based on the constraints

**And** each team of 3 students should avoid having only one member from any critical attribute group

**And** each team of 4 students should avoid having only one member from any critical attribute group

**Scenario:** Verifying that each team after shuffling has no duplicate members

**Given** teams of 3 or 4 students have been generated that meet diversity constraints

**When** the teacher shuffles the teams

**Then** each team should contain unique students

**And** no student should appear in multiple teams

**Scenario:** Validating that the shuffled teams remain distinct from the original

**Given** teams of 3 or 4 students have been generated that meet diversity constraints

**When** the teacher shuffles the teams

**Then** the new teams should be different from the initial team setup

**And** all teams should still meet diversity constraints based on the defined factors

**Feature:** View Change Summary

As a professor,  
So that I can keep track of modifications over time,  
I want to view a summary of changes made to a form.

**Background:**

**Given** I am on the form management page  
**And** there is an existing form with a history of modifications

**Scenario:** Professor views form change summary

**When** I select an existing form  
**And** I click on "View Change History"  
**Then** I should see a list of all modifications made to the form  
**And** each modification should display the date, time, **and** type of change

**Scenario:** Professor sees recent changes at the top of the summary list

**Given** I am viewing the change history of a form  
**Then** the modifications should be listed in chronological order with the most recent changes at the top

**Scenario:** Professor views detailed information for each modification

**Given** I am viewing the change history of a form  
**Then** each modification should display the date **and** time of the change  
**And** each modification should include the type of change (e.g., "added field," "updated field," "deleted field")  
**And** each modification should specify the affected fields or elements

**Scenario:** Professor sees empty change summary for a newly created form

**Given** I have selected a newly created form with no modifications  
**When** I click on "View Change History"  
**Then** I should see a message indicating that there are no modifications yet

**Feature:** Team Generation Button

As a professor,  
So that I can automatically create balanced teams based on collected data,  
I want to click a "Generate Teams" button that uses the defined criteria **and** weightages.

**Background:**

**Given** I am logged in as a professor  
**And** I have defined criteria **and** weightages for team generation  
**And** the necessary data is available in the database

**Scenario:** Generate teams

**When** I visit the team management page  
**Then** I should see a "Generate Teams" button  
**When** I click the "Generate Teams" button  
**Then** the system should retrieve data from the database  
**And** I should see a message "Teams have been successfully generated."  
**And** the generated teams should be displayed on the page

**Scenario:** Ensure data retrieval success

**Given** the database contains valid data for team generation  
**When** I click the "Generate Teams" button  
**Then** the system should retrieve the data without errors

**Scenario:** Handle empty database

**Given** the database does not contain sufficient data for team generation  
**When** I click the "Generate Teams" button  
**Then** I should see an error message "Not enough data to generate teams."

**Feature:** Team Data Export

As a professor,

So that I can use team information in other systems (review, print, or distribute team assignments),  
I want to export generated teams in various formats (Excel, CSV, PDF).

**Background:**

**Given** I am logged in as a professor  
**And** I have generated teams available for export

**Scenario:** Export teams in Excel format

**When** I visit the team management page  
**And** I select the "Export" option  
**And** I choose "Excel" as the export format  
**When** I click the "Export" button  
**Then** I should see a message "Teams exported successfully in Excel format."

**Scenario:** Export teams in CSV format

**When** I select "CSV" as the export format  
**When** I click the "Export" button  
**Then** I should see a message "Teams exported successfully in CSV format."

**Scenario:** Export teams in PDF format

**When** I select "PDF" as the export format  
**When** I click the "Export" button  
**Then** I should see a message "Teams exported successfully in PDF format."

**Scenario:** Validate export without generated teams

**Given** there are no generated teams  
**When** I try to export the teams  
**Then** I should see an error message "No teams available for export."

**Feature:** Programming Proficiency Implementation Part-2

**Scenario:** System ensures varied programming proficiency in teams

**Given** the submission period has closed,  
**When** the system processes student data for team formation,  
**Then** each team should include students with different levels of programming proficiency.

**Scenario:** System allocates remaining unassigned students based on proficiency

**Given** there are students who are not yet allocated to any team,  
**When** the system distributes these remaining students to teams,  
**Then** students should be placed in teams where they add diversity to programming proficiency,  
**And** no team should have students with only the highest or lowest proficiency levels.

**Scenario:** Team formation includes proficiency diversity across all teams

**Given** a group of students with a range of programming proficiency scores,  
**When** the system forms teams,  
**Then** each team should have a balanced mix of high, medium, **and** low proficiency students,  
**And** the overall average proficiency score across teams should be relatively balanced.

**Feature:** Save Form Responses as Draft

As a student,  
So that I can complete the form at my convenience,  
I want the system to save my form responses as a draft if I decide not to submit it immediately.

**Scenario:** Student saves form as draft **and** returns to dashboard

**Given** I am filling out a form,  
**When** I decide not to complete the form responses at that time,  
**And** I choose to save my progress,  
**Then** my current responses should be saved as a draft,  
**And** I should be navigated back to my dashboard,  
**And** I should be able to access the draft form later from my dashboard to continue **and** submit the responses.

**Feature:** Reopen Form

As a professor,  
So that students can continue submitting or editing their responses,  
I want to reopen a form after editing.  
Restrict editing the form **when** it is open for students.

**Background:**

**Given** I am logged in as a professor  
**And** I have a closed form named "Feedback Form"

**Scenario:** Reopen a closed form

**When** I visit the form page for "Feedback Form"  
**Then** I should see a "Reopen Form" button  
**When** I click the "Reopen Form" button  
**Then** I should see "Form was successfully reopened."  
**And** the form "Feedback Form" should be published

**Scenario:** Student can see reopened form

**Given** the form "Feedback Form" is reopened  
**And** there is a student with access to "Feedback Form"  
**When** the student visits their dashboard  
**Then** they should see "Feedback Form" in their list of available forms

**Feature:** Block Edit While Form is Published

As a professor,  
So that I can avoid discrepancies in form responses,  
I want to be constrained from editing a form **when** it is open to students.

**Background:**

**Given** I am logged in as a professor  
**And** I have a closed form named "Feedback Form"

**Scenario:** Restrict editing an open form

**Given** the form "Feedback Form" is reopened  
**When** I visit the edit page for "Feedback Form"  
**Then** I should see "This form is currently open and cannot be edited."  
**And** I should not see the form edit fields

## SPRINT BACKLOG :

### Sprint Goal:

The goal for this sprint is to finalize and thoroughly test the backend logic that supports the team formation functionalities established in previous sprints. We will develop a comprehensive wrapper that integrates all backend components. Additionally, a mechanism will be introduced in the professor interface to allow the automatic generation of student teams with a simple click, utilizing the predefined criteria and attributes. By the end of this sprint, the backend will be fully functional and tested, providing a robust foundation for the team formation process, and the automatic team generation feature will be ready for user interaction.

### Stories pulled into the sprint, assigned developers and story points:

Team Members	Assigned User Stories	Story Points and Time Estimates
Hitha Magadi Vijayanand	Gender Logic - part 2	2 points (up to 1 day)
	Detailed Response	2 points (up to 1 day)
Phani Jyothi Kurada	Ethnicity Logic - part 2	3 points (up to 2 days)
	Update Existing Form Responses On Editing The Form	2 points (up to 1 day)
Ramya Reddy Gotika	Block Submissions After Deadline	2 points (up to 1 day)
Raja Karthik Vobugari	Swap Team Members to Ensure Balanced Teams While Maintaining Gender and Ethnicity Constraints	3 points (up to 2 days)
	View Change Summary	2 points (up to 1 day)
Sahithi Duppati	Team Generation Button	3 points (up to 2 days)
	Team Data Export	2 points (up to 1 day)
Sai Jaideep Reddy Mure	Programming Proficiency Logic - part 2	3 points (up to 2 days)
	Save Form Responses as Draft	2 points (up to 1 day)
Uday Kumar Reddy Mettukuru	Reopen Form	2 points (up to 1 day)
	Block Edit While Form is Published	2 points (up to 1 day)



---

## PROJECT RESOURCES:

Heroku : <https://team-formation-a0e77914b192.herokuapp.com>

GitHub Repo : <https://github.com/phanijyothi11-tamu/team-formation.git>

Project Management page : <https://github.com/users/phanijyothi11-tamu/projects/1>

Slack workspace : <https://app.slack.com/client/T07PN8XD8D7/C07P6NWELCD>

---