PROJECT: TEAM FORMATION
SPRINT-3 Retrospective

SPRINT-3 Schedule :

Sprint 3 Planning : 21st Oct to 25th Oct
Sprint 3 : 28th Oct to 1st Nov

SPRINT-3 Goal :

The goal for this sprint was to finalize and rigorously test the backend logic for team formation functionalities developed in previous sprints. We aimed to create a comprehensive wrapper to integrate all backend components. Additionally, we focused on enhancing the professor interface by introducing an automated team generation feature. This feature allows professors to form student teams with a single click based on predefined criteria and attributes. By the end of the sprint, our objective was to have a fully functional backend, enabling effective user interaction with the new team generation feature.

Team and Member Contributions :
"Team formation" team consists of 8 members. The tabular representation shows the roles, efforts and fair share effort (%).

Team Member	Role	Tasks Completed	Story points (Effort)	Total no. of hours worked	Fair share percentage(%)
Ramya G	Product Owner	2	5	15	12.5%
Rithika S	Scrum Master	2	5	15	12.5%
Uday M	Developer	6	13	21	12.5%
Hitha MV	Developer	3	6	9	12.5%
Sai Jaideep Reddy Mure	Developer	2	5	9	12.5%
Phani Jyothi K	Developer	2	5	9	12.5%
Raja Karthik V	Developer	2	5	9	12.5%
Sahithi Duppati	Developer	2	5	9	12.5%

SPRINT-3 Achievements :

➔ View Responses for a Form: Professors can now view student responses once the students fill the published form.

➔ Update Existing Form Responses on Editing the Form: When a form structure is edited, all existing responses are updated to reflect the new form structure.

➔ Block Submissions After Deadline: The system now blocks students from submitting forms after the specified deadline, ensuring fairness in submission timing.

➔ View Change Summary: Professors can view a summary of all changes made to a form.

➔ Team Generation Button: A "Generate Teams" button has been implemented that allows professors to automatically create teams based on predefined criteria and weightages.

➔ Team Data Export: Professors can now export generated team data in various formats, including Excel, CSV, and PDF, for further review or distribution.

➔ Save Form Responses as Draft: Students can save their form responses as drafts, allowing them to complete and submit the form at a later time.

➔ Reopen Form: Professors can reopen forms after editing, allowing students to continue submitting or editing their responses.

➔ Block Edit While Form is Published: The system now restricts professors from editing a form while it is open to students, preventing discrepancies in form responses.

➔ Unify Deadline Over All Time Zones: The system has been modified to handle deadlines across different time zones, ensuring deadlines are displayed correctly according to the user's local time zone.

➔ Code Refactoring: The forms controller has been refactored to reduce the number of functions, and the create method in the attributes controller has been optimized to reduce code complexity, improving maintainability.

➔ Block Edit While Form is Published: The restriction on editing forms when they are published and open to students has been further reinforced to prevent accidental changes.

SPRINT-3 Backlog Items and Status:

Sprint-3 Backlog Items	Status
Unify Deadline Over All Time Zones	+
Code Refactoring	+
Block Edit While Form is Published	+
Updation of gender implementation	+
Block Submissions After Deadline	*

Populate Teams by gender, ethnicity, Programming skills and Swap Team Members	*

- **Block Submissions After Deadline**(*) : Initially, it was assigned to the product owner (PO), but later it was reassigned to a developer due to the developer's higher bandwidth and availability.

- **Populate Teams by gender, ethnicity, Programming skills and Swap Team Members(*)** : In Sprint 3, we had to change our team formation logic due to the complexity of the swapping step, which required identifying similar students across teams and swapping them without disrupting balance. This swap logic was intended to add variability, but it proved too complex. To address this, we simplified the process by initially assigning students based on gender and ethnicity and introducing randomness by selecting students randomly from these groups. However, the teams generated using this approach were not balanced with respect to proficiency levels, so we had to drop this logic and refine the existing approach to meet the client's requirements for Sprint 4.

Design Diagrams for SPRINT-3 :

Student Details Form Submission

Edit Submission

First name :

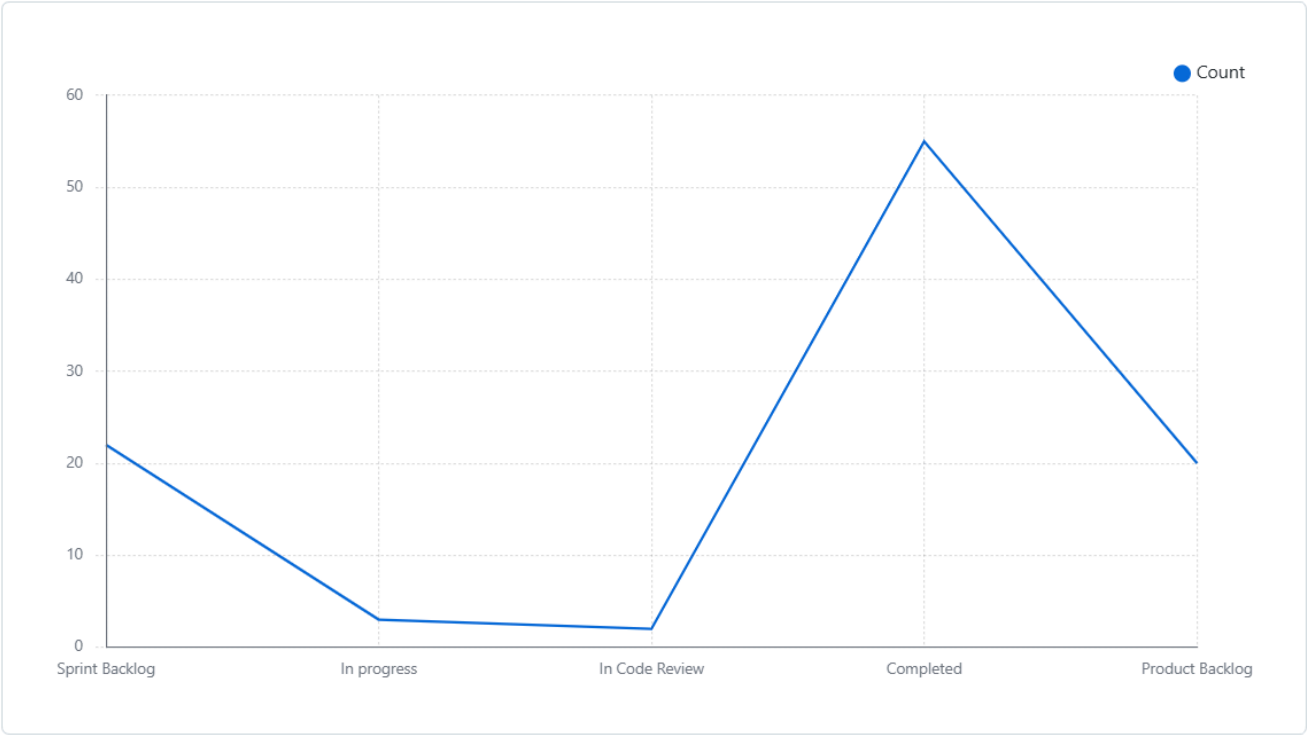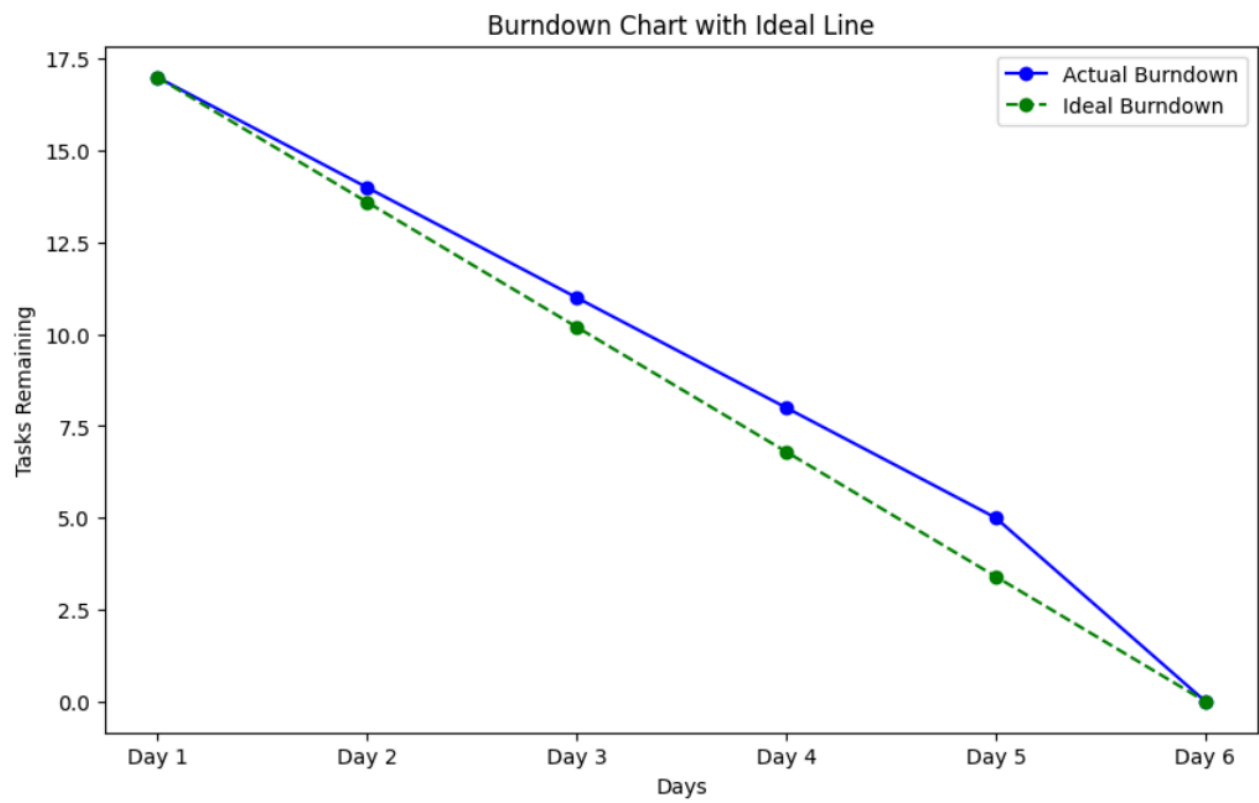Middle name:

Last name:

UIN :

Email:

list of pref.

.

.

Submit

Burndown Chart :

Burndown Chart with Ideal Line

Evaluations of Code & Test Quality :

All Files (97.56% covered at 7.02 hits/line)

24 files in total.
616 relevant lines, 601 lines covered and 15 lines missed. (97.56%)

Search:

File	% covered ▲	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/forms_controller.rb	89.25 %	235	93	83	10	8.75
app/controllers/concerns/form_uploading.rb	95.00 %	72	40	38	2	5.80
app/controllers/form_responses_controller.rb	96.92 %	142	65	63	2	3.49
app/controllers/concerns/populate_teams_based_on_gender.rb	99.08 %	181	109	108	1	5.17
app/controllers/application_controller.rb	100.00 %	37	13	13	0	64.00
app/controllers/attributes_controller.rb	100.00 %	150	64	64	0	8.00
app/controllers/concerns/export_teams.rb	100.00 %	38	24	24	0	2.21
app/controllers/concerns/form_deadline_management.rb	100.00 %	17	9	9	0	2.78
app/controllers/concerns/form_publishing.rb	100.00 %	42	24	24	0	4.21
app/controllers/concerns/generate_teams.rb	100.00 %	48	26	26	0	3.00
app/controllers/sessions_controller.rb	100.00 %	27	16	16	0	3.81
app/controllers/students_controller.rb	100.00 %	47	26	26	0	3.04
app/controllers/users_controller.rb	100.00 %	12	3	3	0	10.67
app/controllers/welcome_controller.rb	100.00 %	16	9	9	0	5.44
app/jobs/check_form_deadlines_job.rb	100.00 %	9	4	4	0	1.00
app/models/application_record.rb	100.00 %	3	2	2	0	2.00
app/models/attribute.rb	100.00 %	54	16	16	0	14.25
app/models/form.rb	100.00 %	59	21	21	0	16.38

Code Cimate: https://codeclimate.com/github/phanijyothi11-tamu/team-formation

Rubocop:

```
Inspecting 126 files
......................................................................................................

126 files inspected, no offenses detected
rithika@DIR-Rural:~/team-formation$
```

bdd & tdd

cucumber scenarios and rspec examples cover all features developed in this iteration.

Rspec:

```
rithika@DIR-Rural:~/team-formation$ bundle exec rspec
WickedPdf.config= is deprecated and will be removed in future versions. Use WickedPdf.configure instead.
.................................................................................................
.................................................................

Finished in 6.47 seconds (files took 6.28 seconds to load)
209 examples, 0 failures

Coverage report generated for RSpec to /home/rithika/team-formation/coverage.
Line Coverage: 97.4% (600 / 616)
rithika@DIR-Rural:~/team-formation$
```

cucumber:

```
    Given I have already published a form                          # features/step_definitions/view_form_responses_steps.rb:6
    And students have submitted their responses                    # features/step_definitions/view_form_responses_steps.rb:15
    When I click on "View Responses" to view responses for a form  # features/step_definitions/view_form_responses_steps.rb:20
    Then I should see a list of student responses                  # features/step_definitions/view_form_responses_steps.rb:25

  Scenario: Professor views detailed student response              # features/view_form_responses.feature:24
    Given I have already published a form                          # features/step_definitions/view_form_responses_steps.rb:6
    And students have submitted their responses                    # features/step_definitions/view_form_responses_steps.rb:15
    When I click on "View Responses" to view responses for a form  # features/step_definitions/view_form_responses_steps.rb:20
    And I select a response and click on "View Details"            # features/step_definitions/view_form_responses_steps.rb:30
    Then I should see the detailed response                        # features/step_definitions/view_form_responses_steps.rb:34

Undefined Scenarios:
cucumber features/form_responses_database.feature:6 # Scenario: Store student responses in the database table
cucumber features/form_responses_database.feature:23 # Scenario: Ineligible student attempts to access the form

45 scenarios (2 undefined, 43 passed)
297 steps (2 skipped, 5 undefined, 290 passed)
0m3.163s

You can implement step definitions for undefined steps with these snippets:

Given('I have uploaded a list of eligible students for the form') do
  pending # Write code here that turns the phrase above into concrete actions
end

When('an eligible student submits their responses') do |table|
  # table is a Cucumber::MultilineArgument::DataTable
  pending # Write code here that turns the phrase above into concrete actions
end

When('an ineligible student attempts to access the form') do
  pending # Write code here that turns the phrase above into concrete actions
end

Then('they should see an error message {string}') do |string|
  pending # Write code here that turns the phrase above into concrete actions
end


  ┌──────────────────────────────────────────────────────────────────────────┐
  │ Share your Cucumber Report with your team at https://reports.cucumber.io   │
  │                                                                            │
  │ Command line option:      --publish                                        │
  │ Environment variable:     CUCUMBER_PUBLISH_ENABLED=true                     │
  │ cucumber.yml:             default: --publish                               │
  │                                                                            │
  │ More information at https://cucumber.io/docs/cucumber/environment-variables/│
  │                                                                            │
  │ To disable this message, specify CUCUMBER_PUBLISH_QUIET=true or use the     │
  │ --publish-quiet option. You can also add this to your cucumber.yml:        │
  │ default: --publish-quiet                                                   │
  └──────────────────────────────────────────────────────────────────────────┘

Coverage report generated for Cucumber, RSpec to /home/rithika/team-formation/coverage.
Line Coverage: 97.56% (601 / 616)
Stopped processing SimpleCov as a previous error not related to SimpleCov has been detected
rithika@DIR-Rural:~/team-formation$
```

Documentation of changes :

Initially the app was deployed on https://team-formation- a0e77914b192.herokuapp.com/ but due to some technical issues the deployment link got changed to this: https://teamformation-c5eaebc1d53b.herokuapp.com/.
.

SPRINT-3 MVP Demo Summary:

- Date: 07 November 2024
- Time: 1:00 PM
- Place: ZACH 424

DISCUSSION POINTS:

1. **Feature Demonstrations:**
   a. Presented and demonstrated the following functionalities:
      i. **Block Submissions After Deadline**: Automatically prevents submissions after the form deadline.
      ii. **Reopen Form:** Ability to reopen the form as needed.
      iii. **Unified Time Zone:** Standardization of time zones across submissions.
      iv. **Save Form Responses as Draft:** Users can save and return to partially completed forms.
      v. **Generate Teams:** Automated team formation.
   b. **Client Feedback:** The client expressed satisfaction with these features, noting that they align well with the project's goals.
2. **Backend Logic Explanation:**
   a. Provided an in-depth explanation of the backend logic supporting each feature, covering implementation details and functionality.
   b. **Client Approval:** The client approved the backend logic and was comfortable with the overall approach.
3. **Priority Stories and App Enhancements for Sprint 4:**
   a. Presented the prioritized stories and planned enhancements for Sprint 4, ensuring alignment with the project timeline and objectives.
   b. **Client Feedback:** The client appreciated the organization of the priority stories. We requested her review of the planned app enhancements, and she plans to provide feedback on their prioritization soon.

NEXT STEPS:
1. Client Review: The client will review the suggested app enhancements and priority stories for Sprint 4 and provide feedback on prioritization.
2. Next Meeting: Scheduled for Monday at 4:15 PM in ZACH 424.
3. Meeting Link: https://meet.google.com/rue-uyno-swr

Project Resources :

Heroku : : https://teamformation-c5eaebc1d53b.herokuapp.com/
GitHub Repo : https://github.com/phanijyothi11-tamu/team-formation
Project Management page : https://github.com/users/phanijyothi11-tamu/projects/1

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