Team Agreement

Team Names: Elsa Merrett, Yanguo Cao, Madhur Jain, Chao li, and David Ohimain

**Contract**

Team Goals:

The aim is to **build a co-creative community knowledge website.** The website will help communities exchange knowledge and share practices about their projects. The website will be where communities planning a project can learn about others like their own. They will find case studies, best practice, and links to more information.

Team Objectives

* Adopt Scrum methodology using Gitlab to deliver a successful web application.
* Break up activities into smaller user stories and assign Git issues around them.
* Ensure that all team members get the opportunity to contribute to front-end and back-end development.
* Ensure we produce quality software using various testing techniques such as writing unit tests wherever applicable.
* Fully demonstrate all acceptance criteria.

Team Member Roles

* Scrum master role will be rotated each week.
* All team members must communicate respectfully and listen to each other's ideas.
* Each team member should contribute working software to Gitlab every week before the Sprint Review.
* Team members should work at least 35 hours a week on the project.
* All team members are responsible for keeping the team updated on individual task progress and for reporting any issues or setbacks they have encountered.
* Each team member will be responsible for reviewing the code of other team members.
* Team members will communicate using the MSc SE - Co-creative Community Knowledge Project team, Team 4 channel on Microsoft Teams.
* Each team member should attend the 11:00am Daily Scrum meeting.
* If a team member is not able to make the pre-agreed meeting time this should be communicated as soon as possible, and the meeting time rearranged to include them if possible.
* Team members should not miss two meetings in a row unless unforeseen/unavoidable circumstances occur (e.g., family emergency, illness etc.)
* All team members should engage in team meetings and discussions.
* The responsibility of developing the agenda and writing minutes for the meeting will be rotated to a different team member each meeting.
* If a team member is unable to complete their assigned work on time, pair programming should be utilized to expediate completion of the task and to help address misunderstandings.
* Merge requests should be reviewed within 24 hours, and constructive comments given on how to improve the code for resubmission if denying the original merge request.
* Where conflicts or disagreements occur, team members should refer to the team agreement for resolution.

Team Norms

* Advanced Agenda- To get most out of the scrum meetings, setting an advance agenda before a scrum can help achieve that.
* Allow Constructive Silence – The scrum leader should create an environment so that team members to be as constructive as they can when they answer after thinking and not saying the first thing that comes into their mind.
* Celebrate Accomplishments – Appreciating each other’s achievements will encourage in creating a healthy environment within the team.

Team Member Evaluation

* Peer evaluations and self-evaluations
* Attendance of each meeting
* Specify something you learned from other members
* Specify something others learned from you

Definition of Ready:

* User stories should be clearly defined.
* User stories should be testable.
* User stories should be accomplishable.
* User stories acceptance criteria should be clearly defined.
* User stories dependencies should be identified.
* Security criteria are identified where appropriate.
* Scalability criteria are identified where appropriate.
* Performance criteria are identified where appropriate.
* User story weight should be estimated and agreed upon by the development team.
* The team member to be assigned to the user story is identified.
* The team understands how to demonstrate the user story.

Definition of Done:

* All code for a specific increment has been produced.
* Code has been commented, merged and run from the current version in source control.
* Code has been peer-reviewed or produced using pair programming.
* Code is free of errors.
* Unit tests have been written and passed.
* Any configuration changes are implemented/documented/communicated.
* Relevant documentation has been produced and/or updated.
* The task has been set to closed.

Additional Technology Used

1. Implement spring jpa which allows developer to write java classes and annotate them. Then spring jpa will convert those classes into data base tables.
   1. Team acceptance here.
      1. Madhur Jain –Accepted
      2. Elsa Merrett – Accepted
      3. Yanguo Cao – Accepted
      4. David Ohimain – Accepted
2. Fetch Api – To interact with the backend from front end, Fetch API is used.
   1. Team acceptance here.
      1. Madhur Jain – Accepted.
      2. Elsa Merrett – Accepted.
      3. Yanguo Cao – Accepted
      4. David Ohimain – Accepted