# Small group project self-assessment

**Team name**: *insert team name here*

The table below contains a list of epics mentioned in the assignment. For each epic that the team produced features for, add a brief explanation outlining how the user can access and use it. You have been provided with some code that incorporates some of the features you need (though you can extend it). This self-assessment will be used to help the marker figure out how to access your system, so try to keep it clear and concise.

Do review the marking scheme for the small group project to understand how the project is assessed. You are reminded that the assignment is **not** to try and implement every single epic.

|  |  |
| --- | --- |
| **Epic** | **Implementation** |
| Epic 1  The web application should have user authentication and authorisation mechanisms. Users should be able to create accounts, log in, and manage their profiles. | *The features in this cell were provided in the scaffolding project.*   * *A home screen gives users the option to sign-up or login (URL/)* * *Users without account, who are not logged in, can sign up as new users (URL/sign\_up)* * *Users with an account can log in (URL/log\_in)* * *Logged in users can log out (URL/log\_out)* * *Logged in users can change their password (URL/password)* * *Logged in users can change their profile (URL/profile)* * *After completing an account related task, logged in users are redirected to a currently blank dashboard (URL/dashboard)* * *If a not-logged in user tries to access pages requiring login, they are redirected to the login page, and then to the page they were after following successful login.* * *A menu is available on all pages while users are logged in.* |
| No further implementation |
| Epic 2  Users can form teams, and invite team members to their team. | * Within the board creation menu, Users are able to add team members should they choose “Team” as the board type. * Users can only add users that have created an account for this application and can not invite them again if they are already part of the team. * After adding, this creates the board for the added user, which will appear in their dashboard when they log in * When viewing the board, they are prompted with an invitation message allowing the user to accept/decline the invitation. * Accepting allows access to the board functions * Declining means they are unable to access the board (as they’re not in the team) * Within the board, the user can add members to the board, in the menu panel of the navigation bar. |
| Epic 3  Each user can create tasks, assign tasks to other team members, and set due dates. | * Within the board, there is a button to create a task. * Clicking on it takes user to a different page to fill out a form to create a task. * The created task will be displayed in the To Do list. By default only the name is displayed * Clicking on the tasks reveals the rest of the information such as description, assigned users and due date. * There are buttons that allow the user to change the name and description and assign users. * A button located next to the task allows the user to move the task to a different list |
| Epic 4  Users provide a dashboard to view and manage assigned tasks, and team tasks. | * Each user will have a main page which displays board. * Each board will store its own set of tasks. * Users are able to create boards * Each board comes with three lists that measure the progress of the tasks. Here it displays the tasks that have been created. |
| Epic 5  Enable users to search, order and filter tasks. These could be based on name, completion status, priority, due date ranges, assigned developer, or team/project. | Not implemented |
| Epic 6  Add a priority system for tasks, allowing users to assign priority levels and filter tasks based on priority. Include automated reminders for high-priority tasks nearing their due dates. | Partially Implemented   * Users are able to assign a priority to tasks, through a dropdown within the task options. * Priorities (LOW, MID, HIGH) * The tasks are then highlighted a certain color in the board based on the priority (green, yellow, red respectively) |
| Epic 7  Introduce task dependencies, enabling users to link tasks and establish dependencies between them. Ensure that the application adjusts task timelines based on dependencies. | Not implemented |
| Epic 8  Integrate a time tracking feature for tasks, allowing users to log time spent on each task. Provide summary reports of time spent on tasks over different periods. | Not implemented |
| Epic 9  Implement an activity log that records user actions within the application, providing an audit trail for accountability. | Not implemented |
| Epic 10  Add a notification system (within the constraints of a Django) to alert users about task assignments and upcoming due dates. | Not implemented |
| Epic 11  Introduce gamification elements to motivate users, such as achievement badges, leaderboards, or a points system based on task completion. | * Each User will have their own achievements. * Each achievement has their own milestones e.g completing the achievement 10,100 times. * A screen that displays the achievements * Users can get achievements based on actions they perform e.g adding/deleting board, creating tasks, “completing” tasks (moving it to completed), logging in * Users can click on a button to view more information about the achievement.   Here it will display the date which a milestone has been achieved |