# Small group project self-assessment

**Team name**: *Gecko*

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 dashboard where the teams and tasks are displayed. Initially both sections of the dashboard are empty with messages giving an overview to the user on the layout of the website.* * *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. | * *Users can see the teams on the left-hand side.* * *Users can create teams using the Create team section on the menu.* * *If a user fails to fulfil the required fields of the form, they will receive a validation message for guidance.* * *In the Create Team form, users can add name, description and select the members of the team by CTRL+Click.* * *When a team is created, invites are sent out to all selected team members (except the team admin). Potential members can choose to either join the team or reject the invite* * *After successful team creation, users are redirected to the dashboard.* |
| Epic 3  Each user can create tasks, assign tasks to other team members, and set due dates. | * *Users can create tasks using the Create Task section on the menu.* * *In the Create Task form, users can select the team of the task and can assign the task to team members, set a due date, set status and priority values.* * *If a user fails to fulfil the required fields of the form, they will receive a validation message for guidance.* * *After successful task creation, users are redirected to the dashboard.* |
| Epic 4  Users provide a dashboard to view and manage assigned tasks, and team tasks. | * *Every team card has a view details section that shows description, admin, and member details as well as the team tasks.* * *Team admins can transfer admin roles, add/remove members and delete the team.* * *Both regular and admin members can leave the teams.* * *The Task section of the dashboard would show the user’s assigned tasks from respective teams.* * *Add task management section of the dashboard, how task editing works, how they are displayed on the tasks section.* |
| 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. | * *Users can search order and filter tasks by name, status, and priority in the Task section of the dashboard.* * *Add more details about how this looks on the front end.* |
| 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. | * *Every task has a priority system with the values of Low, Medium, and High.* * *Time sensitivity has not been implemented for the priority system at the moment.* |
| 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. | Not implemented |