# Small group project self-assessment

**Team name**: *Impala*

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.* |
| We added a rest password button when you click log in. This allows you to input a username which will send an email to the user to a page where they can input their password changes and submit. |
| Epic 2  Users can form teams, and invite team members to their team. | An Invite team member screen gives you a pop up with a space to write a user’s name.  You write the users username and an email is sent to them with a link.  The link takes them to the team dashboard. |
| Epic 3  Each user can create tasks, assign tasks to other team members, and set due dates. | In the team dashboard, you can click the button create task.  That takes you to a page which has a form which you can fill out.  That consists of assigning members, assign jelly points, assign due date and set priority.  The back button on the page redirects you to team dashboard |
| Epic 4  Users provide a dashboard to view and manage assigned tasks, and team tasks. | Clicking on the task name will give you a pop up that will give you three buttons.  You can delete the task, edit task and go back to dashboard  Edit task takes you to a popup and clicking back redirects you to the team dashboard.  Back redirects you to the team dashboard.  Delete task redirects you to the team dashboard. |
| 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. | The filtering is in my tasks button in the navigation bar as well as the team dashboard.  You have select priority which allows you to choose between High, Medium and Low priority  You have a start date and end date which allows you to choose dates.  You have a search bar to allow you to filter team names. |
| 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. | * During creation or editing of tasks, users are able to assign priority levels of tasks. * The priority is displayed while displaying tasks. * In Team Dashboard, you are able to filter to view tasks based on priority. |
| 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. | * There is an automatic time tracking feature for tasks. * The duration between creation and completion of tasks is calculated and displayed. |
| 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 task is assigned “jelly points” (user story points) by the user during creation of tasks. * Users are able to edit the number of jelly points if necessary. * Once a user has completed a task, the jelly points of the task are stored in a total and displayed within the user’s dashboard. |