Small group project self-assessment

**Team name**: Kangaroo

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. | * Logged in users can use the dashboard to create a team (URL/dashboard) * The team creation popup menu includes the team’s name, description, and members to invite * All team members can see the team’s details from a dropdown menu * The team creator can invite more members with an invite message, and remove members from this menu * There is an autocomplete search feature to find users that can be added to a team * The team creator can delete the team (URL/delete\_team/<int:team\_id>) |
| Epic 3  Each user can create tasks, assign tasks to other team members, and set due dates. | * Logged in users can use the dashboard to create tasks (URL/dashboard) * The task creation popup menu includes the task’s name, description, dependencies, priority level, due date and due time * Tasks can be edit (URL/task\_edit/<int:pk>) * If the inputs are incorrect, the user is redirected to a page specifying the errors (URL/task\_create) |
| Epic 4  Users provide a dashboard to view and manage assigned tasks, and team tasks. | * Logged in users can manage lanes, tasks, and teams from the dashboard (URL/dashboard) * Lanes can be selected and renamed. Press “enter” to confirm the name change * Lanes can have their order changed * Lanes can be deleted (URL/lane\_delete/<int:lane\_id>) * Tasks can be assigned to users and moved between lanes * Tasks can be deleted (URL/task\_delete/<int:pk>) |
| 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. | * Logged in users can search and order tasks (URL/task\_search) * The table includes the task’s name, description, due date, priority level, and assigned team * Users can order by due date and priority level * Users can search tasks by their name |
| 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. | * Logged in users can assign priority levels to a task from the task create popup (URL/dashboard) * The priority levels are low, medium, and high * When a task is nearing its deadline, all users in that task’s team are notified * The notification can be found on the navbar |
| 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. | * When creating a task, the user can set dependencies on tasks in the team * When editing a task, the user can set the dependencies on tasks in the team excluding itself * A task with dependencies has a grey task name on the dashboard, while other tasks are in black * Tasks with dependencies have no functionality restrictions; the distinction is purely visual |
| 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. | * A button marked with a bell icon in the navbar allows logged in users to view their notifications in a popup * Users are shown their notifications in order of the most recent notification first * When a user is assigned a task, that user is notified * When a task reaches 5 days before its deadline, all users in that task’s team are notified * As a deadline approaches, the notification for that task gets refreshed and placed at the top of the user’s notifications tab * If a deadline is approaching, then is edited to a date that is after 5 days from the current date, the notification is deleted. * When a user is invited to a team, they are notified and can accept or reject the invitation via the notifications tab |
| Epic 11  Introduce gamification elements to motivate users, such as achievement badges, leaderboards, or a points system based on task completion. | Not implemented |