py pip -m install Flask

To run flask : flask run

$ pip install Flask-SQLAlchemy

[Trello link](https://trello.com/invite/b/4YXjEXPP/ATTI87c22a234fa98424931e5a8d71e11f27C992D49E/team-24)

**Sprint Planner 1 :**

***Part 1 : Our Sprint Goals:***

* Agreed upon submission deadline: 13/09/23
* By the agreed upon submission deadline, we shall extend on the existing product by adding/adapting to these features:

1. Implementing error messages to the log-in system
2. Extend with the drag and drop for tasks, move the tasks across the columns.
3. Add tags to the task to sort them across the 8 categories given, they should be colour coded
4. Differentiation between Admin and User accounts (using inheritance) : a couple of admin accounts and user accounts.
   1. All user accounts (abstract class) , Admin account , user account
5. Implementing a database so that the same content is accessible, where changes are synced across different user sessions.
6. Implement a user account page where user accounts can update their password ~~s and add the average time spent~~
7. Implement an admin account page where admin can control who to add as users, allow for them to change passwords, and see analytics.
8. Non-functional user requirements: Security - making sure you follow through with security, making sure users accounts are secure:
   * + Keep track of usernames and passwords so they cannot easily be changed by someone who is unauthorised. Keep a log of how many times a password has been changed, if it has been more than 3 times (within 5 minutes, if continuous incorrect attempts then 1 hour), stop them from being able to change it any more.
     + Passwords stored securely
     + Authentication: Make sure you are who you say you are
     + Authorisation: Make sure you have the permission to be doing certain tasks (i.e. admin tasks)
9. Start implementing analytics features :
   1. Using user’s input of their estimate hours put into individual tasks and plotting them
   2. Plotting the overall user input

**Sprint Items Priority** (See sprint backlog for specific user stories)**:**

1. Add tags to the task to sort them across the 8 categories given, they should be colour coded
2. Implementing a database so that the same content is accessible, where changes are synced across different user sessions.
3. Extend with the drag and drop for tasks, move the tasks across the columns.

***Part 2 : Sprint Delegation:***

Allocation planning

| Sprint backlog ID | Task | Assigned team member | Agreed deadline |
| --- | --- | --- | --- |
|  | * Differentiation between Admin(a couple of accounts) and User accounts (using inheritance) - don't need a separate proxy owner + admin * Implementing error messages to the log-in system | Milni | 2/10/23 |
|  | Adding drag and drop functionality between the statuses of the sprint backlog. Also, adding functionality to move tasks from the product to the sprint backlog.  Implementing a menu with links to a user page and any extra pages which may be needed. | Carissa | 2/10/23 |
|  | Implement a user account page where user accounts can update their passwords.  Non-functional user requirements: Security - making sure you follow through with security, making sure users accounts are secure: | Kirtan | 2/10/23 |
|  | Implementing a database so that the same content is accessible, where changes are synced across different user sessions.  Change the tasks from being hard coded in, to incorporate what is in the database. | Nilly | 2/10/23 |
|  | Adding colour-coded tags to the tasks (separating them into 8 different categories)  Make it so that the tasks can be edited after they are written | Erin | 2/10/23 |
|  | Sprint Backlog + Retrospective + add extra risks identified below  Instructions page | Max | 2/10/23 |
|  | Analytics: Using user’s input of their estimate hours put into individual tasks and plotting them | Chaitsee | 2/10/23 |

**Additional risks identified during Sprint 2**

| **ID** | **Date Raised** | **Risk Description** | **Likelihood of the risk occurring** | **Impact if the risk occurs** | **Severity Rating** *based on impact & likelihood* | **Owner + Monitoring Strategy**  *Person who will manage the risk* | **Mitigating action** | **Contingency Plans** (if risk has already happened) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 26/09/2023 | Addition of new members leading to miscommunication and or issues with the dynamics of adapting to a new team environment | Medium | Very High | High | To ensure that the risk of adding new team members does not lead to miscommunication and disruptions in team dynamics, the team must establish a clear workflow for new members to adapt to. The team members and leaders should proactively address any issues related to miscommunication and team dynamics, and aim to promote an efficient work atmosphere. | Team members should look to the team lead and owner if there are misconceptions and disagreements with the workflow and/or direction of project scope. | Ensure that if ever there are any feelings leading towards disagreements which may harm the dynamics of the team work environment, communicate ahead of time with the team leader proactively to allow the leader to accommodate and adjust the environment to ensure that all members are comfortable. |
| 24 | 30/09/2023 | Inadequate allocation of resources regarding time, personnel (skill), that may hinder the successful deliverance of the project | Medium | High | Medium | Conduct a comprehensive resource assessment at the project’s initiation stage or at the beginning of each sprint. This should involve collaboration amongst the team to ensure each member is not only capable of their allocated task, but also satisfied. Team members should also cross-train each other to allow for flexibility while compensating for resource shortages. | The team lead should conduct a thorough resource assessment during the process of planning the project. There must be metrics set up to regularly monitor and track resource utilisation, encouraging cross-training among team members to enhance flexibility. | Ensure during the allocation process that tasks being allocated to team members are sufficiently understood by them. In the instance that the team is unable to successfully flex between tasks, ensure that a downsizing of scope aligns with the minimal viable product for the customer. |

## 

Projects NFR:

* Security
* Usability

Overlooked NFRs:

* Portability
* Compatibility
* Performance efficiency
* Maintainability
* Reliability

How will we manage:

* Security: implement secure log in
* Usability: Implement operable UI
* Compatibility: Implement a database
* Reliability: Recovery procedures
* Maintainability: Continuously modify the code
* Portability: Use on different devices

How will we verify:

* Security: Users are not able to access the system in any way without correct login details
* Compatibility: Information stored on one device will be seen on another device
* Portability: Users will be able to use on a variety of devices
* Performance efficiency: Users will not experience long loading times
* Usability: Users from outside of the development team are able to use the system intuitively
* Reliability: Users will be able to complete tasks without breaking the program
* Maintainability: We will be able to continue modifying the code without it breaking

Confirming you are admin while making big changes:

Eg; making someone an admin