****

**Team 31 –Sprint 2 Retrospective**

Pooja Tewari, Shivangi Chand, Siddharth Dhar, Sripath Mishra

What went well?

**General :**

The second sprint went really well in comparison to the first sprint. Our team meetings were very productive, whether they were general meetings or programming meetings we were able to accomplish our agenda for that particular meeting. We first began with finishing up what was not completed in our first sprint. The backend and frontend were linked for the initial user stories that dealt with authentication and then we completed the user stories that dealt with being able to create and navigate to notebooks, making them personal or private. Then we started on the remaining user stories which deal with being able to add pages, navigate to them, adding text to our notebooks and being able to change the text accordingly. We also were able to successfully implement the search bar. Proper testing was also done for all of these user stories. All in all, it was a relatively successful sprint and both the backend and the frontend teams worked timely and hand in hand.

**User​ ​Story​ ​5:**

As a user, I would like to be able to open any page of my notebook

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Selecting the desired notebook to open | 2 | Shivangi |
| 2 | Fetching data from the database | 2 | Shivangi |
| 3 | Projecting notes on the screen | 3 | Sripath |
| 4 | Testing and Quality Assurance. | 2 | Sidharth |

**Completed:** All the parts of this user story was completed in a swift manner. Due to informative and engaging meetings the team was able to quickly write up the code both for the front and the backend. Rigorous testing was done both by using unit test cases and integration test cases in the backend. Frontend manual testing was also done by the team.

**User​ ​Story​ ​9:**

As a user, I would like to be able to insert text in the notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Fetch data from the database | 3 | Shivangi |
| 2 | Permit the user to insert text in the notebook | 2 | Shivangi |
| 3 | Save the changes | 2 | Sripath |
| 4 | Unit Testing: Test Input | 3 | Sidharth |

**Completed:** The backend was handled well along with the database. Everything was tested very properly making sure that the inserted text was being updated and displayed in the notebooks. Previous research over rich text editors and storing files was very helpful. The backend and the frontend team used an creative method to store the notebook data.

**User​ ​Story​ ​10:**

As a user, I would like to be able to update text in the notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Fetch data from the database | 3 | Shivangi |
| 2 | Permit the user to update text in the notebook | 2 | Shivangi |
| 3 | Save the changes | 2 | Sripath |
| 4 | Unit Testing: Test Input updation | 3 | Pooja |

**Completed**: The backend was handled well here as the communication to and from the database worked well. The data was being displayed clearly on the frontend as well. Testing was also well performed to make sure that any changes were being saved.

**User​ ​Story​ ​11:**

As a user, I would like to be able to delete text in the notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Fetch data from the database | 3 | Shivangi |
| 2 | Permit the user to delete text in the notebook | 2 | Shivangi |
| 3 | Save the changes | 2 | Sripath |
| 4 | Unit Testing: Test Input deletion | 3 | Pooja |

**Completed:** The communication to and from the database key here. Any deletion that took place was clearly being updated in the database and being clearly displayed as intended on the frontend as well. Testing was also well performed to make sure that any deletion changes were being saved.

**User Story 12:**

As a user, I would like to be able to beautify text in the notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Permitting the user to bold the text | 3 | Siddharth |
| 2 | Permitting the user to underline the text. | 2 | Sripath |
| 3 | Permitting the user to change the text to italics | 3 | Sripath |
| 4 | Permitting the user to alter the text size | 2 | Shivangi |
| 5 | Unit Testing: Test Input beautification | 3 | Pooja |

**Completed:**

All the parts of this user story were created using an API for being able to beautify the text. The API was integrated properly into the web app and everything was being saved properly. Testing was done in a way to make sure all parts work properly and were being saved in the database.

**User​ ​Story​ ​25:**

As a user, I would like to be able to search other public notebooks by their titles, owners, etc.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create backend route for searching notebooks. | 2 | Sripath |
| 2 | Allow the route to search a notebook by owner, and title of the notebook. | 2 | Sripath |
| 3 | Create UI for the search notebook | 3 | Siddharth |
| 4 | Link the frontend and the backend | 3 | Shivangi |
| 5 | Unit testing and integration testing. | 3 | Pooja |

**Completed:**

The backend was quick to implement two routes to search for notebook by the name and the owner of the notebook. It was unit tested and deployed. The frontend linking was also quick. More focus was given to functionality as compared to a beautiful UI. Later implementation was required on the backend side to prevent users to search private notebook owned by other users. It was done efficiently.

**User​ ​Story​ ​4:**

As a user, I would like to be able to easily navigate to a personal Notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Selecting the desired notebook to open | 2 | Siddharth |
| 2 | Fetching data from the database | 2 | Siddharth |
| 3 | Projecting notes on the screen | 3 | Siddharth |
| 4 | Testing and Quality Assurance. | 2 | Sripath |

**Completed:** All the tasks for this user stories were completed quickly as the backend had been prepared well and tested well. The frontend also did a great job in how the notebooks were being displayed on the dashboard. Proper testing was again done to make sure all the notebooks are being fetched from the database and the correct information is being displayed.

**User​ ​Story​ ​7:**

As a user, I would like to be able to create new notebooks.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create​ a new notebook​ ​in​ the ​database | 2 | Sripath |
| 2 | Create​ ​notebook​ ​model in the backend | 2 | Sripath |
| 3 | Allow the logged in user to be able to access the new notebook | 3 | Sripath |
| 4 | Create the UI for a new notebook | 4 | Siddharth |
| 5 | Save the changes in the database | 2 | Sripath |
| 6 | Testing (unit and integration) | 2 | Shivangi |

**Completed:**

The backend had done a good job making sure that notebooks were being stored in the database properly and then also being fetched properly. The frontend UI was also very important for this part as the notebooks had to be displayed well on the dashboard. Testing was done to make sure that the notebooks are being created and are being displayed as to how the user story intends.

**User​ ​Story​ ​8:**

As a user, I would like to be able to create new pages in any notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create a new page in the database | 2 | Shivangi |
| 2 | Create the new page model in the database | 2 | Pooja |
| 3 | Allow the user to be able to create and access the new page in a notebook | 3 | Shivangi |
| 4 | Create the UI for the new page | 3 | Pooja |
| 5 | Unit​ ​test-​ ​correct​ ​and​ ​incorrect​ ​input, the user is authenticated | 3 | Pooja |

**Completed:**

Both the frontend and backend had to spend time working on making sure that the linking of both is done properly. The database was key in this as it was important to make sure the new page was being stored in the database properly and then also being fetched properly. The frontend UI was also very important for this part. Then a lot of testing was also performed to make sure that the user story works properly.

**User Story 24:**

As a user, I would like to be able to make my notebooks private.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create UI for making private/ public | 3 | Siddharth |
| 2 | Backend for private/ public | 2 | Sripath |
| 3 | Create Database model for private/public | 3 | Pooja |
| 4 | Unit Test: Input testing | 2 | Pooja |

**Completed:**

The front end had to create a UI and it was created properly and completed on time. The backend also did a good job in implementing the feature and also adding that to the database. Upon linking the frontend and the backend, testing was performed to make sure the feature worked properly.

What did not go well?

**General:**

There were small errors that occurred in completing the user stories below. For the user story 26, what went wrong was that the user can only like his/her own notebooks and not other users’ notebooks. That is something that does not fully complete the idea of the user story and we should have been more careful with testing that as a team. The same thing goes for the subscription feature which was not finished to the whole.

**User​ ​Story​ ​26:**

As a user, I would like to be able to like, dislike and linear comment about a notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create a backend route for getting and updating likes, dislikes and comment for a notebook. | 2 | Sripath |
| 3 | Create UI for the comment, likes and dislike about a notebook | 3 | Siddharth |
| 4 | Link the frontend and the backend | 3 | Shivangi |
| 5 | Unit testing and integration testing. | 3 | Pooja |

**Not Completed:** The user can only like his/her own notebooks and not another users’ notebook. The main idea of the user story was to be able to implement like, dislike, and comment on another’s notebook. This bug was previously absent and has to be investigated. The backend routes are unit tested. Therefore, it is more probable for the bug to be in the front end implementation.

**User​ ​Story​ ​27:**

As a user, I would like to be able to subscribe to another user or any updates on a notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Create a backend route for subscribing to a notebook and to send notifications to the user upon request. | 2 | Sripath |
| 3 | Create UI for the subscribing the notebook and to view the notification. | 3 | Siddharth |
| 4 | Link the frontend and the backend | 3 | Shivangi |
| 5 | Unit testing and integration testing. | 3 | Pooja |

**Not Completed:**

This route was also partially incomplete. The user was able to successfully subscribe to a notebook which was confirmed by the database tables. When The notebook was updated the notification column for that user was updated. The incomplete part was the frontend implementation to call the routes, get the notification and then to display it to the user.

How can we Improve?

As a team, we have done a much better job at being honest with each other in terms of our workload in other classes and how much time we could have honestly committed to the project. What worked better for us this time around was just accepting that we did not do well on our first sprint and the best way to begin this second sprint would be to just initially finish everything that was left incomplete from the first sprint. We can definitely get better towards dividing our time better in the sense that does a little bit at a time instead of large amounts of work over large amounts of time in one day. It would be better to do a little bit each day. We can also work more towards doing our work together all the time as we achieve our best results then. It is better to work altogether because then there is no space for miscommunication.

We can also improve more on following the planning document a little better. We made a mistake with our user story 26 and 27 in the sense that we did not realize that the like and dislike feature was only working for the self. We could not go like and dislike other users’ notebooks. That is something that can be done only when subscribing works properly and it was not working properly. So next time we will focus on making sure that our planning document makes sense in a logical order and also we will make sure user stories are working completely in all ways.

Another issue that we faced was with one of our team members falling sick and getting diagnosed with pneumonia for 2 weeks. Due to her illness, there was a slight issue with time as she could not attend the group programming meetings. We can plan better for such emergency situations because illness can happen to anyone. One way to deal with it would be to prepare an action plan in case something like this were to occur where a member of the team is in an emergency situation where they are unable to give a large contribution. In such situations, it would be best that the rest of the teams organizes work in a way where even if the schedule is pushed back we can still accomplish our overall goals.

One more thing that we can work on better is having good communication even on our slack channel. Though the best way that our team communicates is in person during meetings, it is also important that online communication is at its best because of course it is difficult to meet almost everyday and sometimes work does need to be completed remotely. In such situations, it is important that our team improves on sending clear and to the point messages on slack which will keep all our member's updates on exactly what one member is doing.

In conclusion, our team needs to make sure that we are staying on track with our user stories and plan in a way where the Thanksgiving break or any other situation does not harm our schedule, which is possible and can be done through communicating properly.