****

**Team 31 –Sprint Retrospective**

Pooja Tewari, Shivangi Chand, Siddharth Dhar, Sripath Mishra

What went well?

**General :**

The final sprint went exceedingly well in comparison to our second sprint. Our team meetings were very productive as we accomplished our agenda for the particular meeting. Moreover, we were also very serious and professional in our approach towards our daily and weekly objectives. In my opinion, we worked on our user stories with devotion and dedication. We were able to finish and test all the user stories and were satisfied with our work at the end of the sprint. Our team meetings were more of an open discussion in which our team members came up with different ideas or solutions to the same problem. These discrepancies or conflict of opinions were necessary to come up with the most optimal solution to the problem. We added a lot of functionality to our project to make it professional. To name a few, permitting users to comment on notebooks, subscribe to users, upload a pdf and write text on pdf. Lastly, the backend and frontend team were also parallel to each other in progress. To conclude, our team had a very professional approach to the final sprint which helped us tackle all the obstacles along the way quite efficiently.

**User​ ​Story​ ​13:**

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

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Fetch data from the database | 2 | Siddharth |
| 2 | Permit the user to insert pdf in the notebook. | 2 | Siddharth |
| 3 | Save the changes | 3 | Sripath |
| 4 | Unit Testing: Test insertion. | 2 | Siddharth |

* **Completed:**

This route was an interesting route as it had to be researched and the implemented. The user was able to upload their pdf and in the background, it was processed and converted to base 64 encoding. This encoding was passed to the backend and was stored in a new column. This story required a heavy processing in the frontend and the backend routes were updated to incorporate the pdf. The two teams ( frontend and backend ) were fast to process the data and enable the user to add new pdf.

**User​ ​Story​ ​14:**

As a user, I would like to be able to update pdf 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 pdf in the notebook | 2 | Shivangi |
| 3 | Save the changes | 2 | Sripath |
| 4 | Unit Testing: Test pdf updation | 3 | Sidharth |

* **Completed:**
* This route was a required to allow the users to replace the previous pdf with a new an updated one. The user was able to upload their updated pdf and in the background, it was processed and converted to base 64 encoding. The text written on the previous pdf was stored and could be written over the new pdf. This story required a heavy processing in the frontend and the backend routes were updated to incorporate the pdf. The two teams ( frontend and backend ) were fast to process the data and enable the user to update the pdf.

**User​ ​Story​ ​15:**

As a user, I would like to be able to delete pdf 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 pdf in the notebook | 2 | Shivangi |
| 3 | Save the changes | 2 | Sripath |
| 4 | Unit Testing: Test pdf deletion | 3 | Pooja |

* **Completed:**
* This route was required to allow the users to delete the previous pdf. The text written on the previous pdf was stored and could be written over any new future pdf. This story requires the ability to completely remove the previous pdf. The two teams ( frontend and backend ) were fast to process the data and enable the user to delete the pdf.

**User​ ​Story​ ​16:**

As a user, I would like to be able to insert text over pdf.

Task​ ​Table:

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

* **Completed:**
* This route was required to allow the users to write text over the pdf. The user was able to write Latin characters over the text from the top. With the help of previous research, we were able to implement the route swiftly and confidently. The two teams ( frontend and backend ) were fast to process the data and enable the user to add the text.

**User Story 17:**

As a user, I would like to be able to update text over pdf.

Task​ ​Table:

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

* **Completed:**
* This route was required to allow the users to update text written over the pdf. The user was able to write Latin characters over the text from the top. By changing the text in the file structure of the pdf one was able to update the text. With the help of the above user story, we were able to implement the route swiftly and confidently. The two teams ( frontend and backend ) were fast to process the data and enable the user to update the text.

**User​ ​Story​ ​18:**

As a user, I would like to be able to delete text over pdf.

Task​ ​Table:

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

* **Completed:**
* This route was required to allow the users to delete text written over the pdf. The user was able to write Latin characters over the text from the top. By changing the text in the file structure of the pdf one was able to update the text. With the help of the above user story, we were able to implement the route swiftly and confidently. The two teams ( frontend and backend ) were fast to process the data and enable the user to delete the text. A route was implemented which allowed us to store all the text written by the user. This allowed us to make the text over pdf independent of the pdf.

**User​ ​Story​ ​19:**

As a user, I would like to be able to save any changes made in the notebook.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Fetch the data from the database | 2 | Sripath |
| 2 | Permit the user to make any updates in the notebook | 3 | Siddharth |
| 3 | Save all changes | 3 | Shivangi |
| 4 | Unit testing: Test input updation | 3 | Pooja |

**Completed:**

* This route was required to allow the users to save all the notes taken by the user. The user was able to write notes and was able to close the website and see his work after reopening the website. The two teams ( frontend and backend ) were fast to process the data and enable the user to save the notes.

**User​ ​Story​ ​20:**

As a user, I would like to be able to edit notebooks after they have been saved.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Fetch data from the database | 2 | Sripath |
| 2 | Permit the user to make changes to a currently saved notebook | 3 | Siddharth |
| 3 | Save the changes | 3 | Shivangi |
| 4 | Unit testing and integration testing. | 3 | Pooja |

* **Completed:**
* This route was required to allow the users to edit all the notes taken by the user. The user was able to edit all notes and the pdf. This allowed full independence to the user and flexibility to change any part of the notebook. The two teams ( frontend and backend ) were fast to process the data and enable the user to update the notebooks.

**User​ ​Story​ ​21:**

As a user, I would like to be able to organize the notebooks in folders.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Permit the user to create a folder | 2 | Sripath |
| 2 | Put the desired notebooks in the folder | 3 | Siddharth |
| 3 | Get rid of occurrences of the notebooks outside of the folder | 3 | Shivangi |
| 4 | Save the changes to the database | 3 | Pooja |

**Completed:**

* This route was required to allow users to organize their notebooks for supporting easy usage. New routes were implemented to allow the same. Frontend code was also required to make sure the user is able to view the contents of the folder with making the least amount of API calls. This allowed customization of notebooks. The two teams ( frontend and backend ) were fast to process the data and enable the user to place notebooks in folders.

**User​ ​Story​ ​22:**

As a user, I would like to be able to share my notebook or make my notebooks public.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Backend implementation to allow the user to make notebooks public. | 2 | Sripath |
| 2 | Backend implementation to allow notebook viewing by link sharing. | 3 | Siddharth |
| 3 | Frontend implementation to create links. | 3 | Shivangi |
| 4 | Unit testing and integration testing. | 3 | Pooja |

**Completed:**

* This route was required to allow users to make their notebooks public or private. This user story enabled us to differentiate ourselves from the competitors. New routes were implemented to allow the same. Frontend code was also required to make sure the user is able to share the notebook. This allowed sharing of notebooks. The two teams ( frontend and backend ) were fast to process the data and enable the user to place notebooks in folders.

**User​ ​Story​ ​28:**

As a user, I would like to be able to edit any notebooks which I have been given access to by the owner.

Task​ ​Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Task​ ​Number | Description | Time | Task​ ​Owner |
| 1 | Permit the owner to give access to his notebook to a user upon request | 2 | Sripath |
| 2 | Allow the user to insert and delete text in the notebook | 3 | Siddharth |
| 3 | Permit the user to beautify text in the notebook | 3 | Shivangi |
| 4 | Unit testing and integration testing. | 3 | Pooja |

**Completed:**

* This route was required to allow users to edit notebooks one has been given access to. This user story enabled us to allow other users to update notebooks. But, it also helps us to prevent any malicious updates on the notebook. Old routes were updated to allow the same. Frontend code was also required to make sure the user edit other’s notebooks. The two teams ( frontend and backend ) were fast to process the data and enable the user to allow other user’s to update their notebooks.

**User​ ​Story​ ​25:**

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 |

* **Completed:**
* This route was previously partially incomplete. The user was able to successfully like, dislike on any notebooks. But, the user was unable to comment on notebooks of other people. The bug was quickly fixed and the testing was made to be more rigorous. The bug was in the backend SQL statement used.

**User​ ​Story​ ​26:**

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 |

**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. The incomplete part was quickly finished and tested.

What did not go well?

**General:**

As far as the user stories are concerned, we completed what was required of us for Sprint 3. All the user stories were completed in the most optimal manner as they could have been. However, our only concern was the Graphical User Interface or the GUI which was not as appealing as it should have been. Moreover, there were minor bugs with our code such as overlapping of text that needs to be addressed and fixed.

How can we Improve?

As a team, we have improved a lot by this sprint. We were able to finish all our user stories completely. What we could have improved more in is the GUI of the overall app. Because of getting so caught up in finishing the user stories. We did not spend as much time focusing on the UI of the app. What we can work on is refining the colors of the overall app and developing a standard theme of Carnet.

The dashboard is what needs to be worked on mostly and can be improved upon. We can add features such as adding a page for being able to edit your profile and a settings page that allows you to customize the dashboard and the app more. Currently, there is nothing really that is offered for the user to be able to do on their profiles.

The organizations of the notebooks can be improved upon as well. Currently, we can create folders, but there is no graphical way to just drag and drop notebooks into folders. So if we can come up with a way to drag and drop different notebooks into a folder graphically and also folders into other folders graphically.

The notebooks also need to be displayed better. Currently, we just have horizontal divs that we are using to display the divs, but we have to improve by actually showing pictures of a notebook to display each of the divs in a 3x3 grid. The buttons also are not really properly displayed, so we can improve by a lot by working on the CSS of buttons and making them look a little more better in appearance.

The notebooks themselves can look better. We can add lines to the editor so that it looks like what an actual notebook looks like. We can allow the user to add the functionality of being able to put margins. Currently, the pdf feature only allows you to add text over the pdf, but we can improve the app so that the user can edit the pdf from anywhere on any line.

In conclusion, there are a lot of improvements that can be made to the functionality and the looks of the app. Currently, we just spent time trying to really get an MVP together. As a team, we have improved a lot on communicating better and did a good job in getting a little done at a time. There is a lot that we learned about teamwork as part of this project, so that was a good experience.