****

**Team 31 – Product Backlog**

Pooja Tewari, Shivangi Chand, Siddharth Dhar, Sripath Mishra, Zachary Moore

**Problem Statement**

Taking notes in class and keeping them organized can be difficult; sharing those notes amongst friends, classmates and any other prospective users and getting feedback can be beneficial to both parties in terms of the quality of the information that is being exchanged. To simplify this, our project aims to develop a web app that would allow students to create and organize notebooks on their personal accounts in addition to allowing other students to subscribe to him/her, allowing them to provide feedback to these notes. Notebooks will be readily available across all web-enabled devices.

**Background Information:**

**Audience**

Students across the globe make and exchange class notes. A learning experience can be enhanced and made easier if a student has access to well-written notebooks with them already. With Carnet, we aim to provide a strong online platform to create and exchange class notes between the students.

**Similar Platforms**

There are several Class Notes management and Note exchanging platforms. For instance, platforms like Google Drive, OneNote, Word that allows students to create class notes and platforms like Course Hero, Slideshare that allows students to exchange class notes/presentation.

**Limitations**

The existing platforms are widely used by students throughout the world, however, they can either make class notes only or only share them on a public platform. With Carnet, we aim to bridge the gap between the two existing facilities. Carnet will not only provide a platform to create and share class notes but also allow its users to comment on other person’s class notes. Additionally, we will allow users to subscribe to other users. We aim to create a high-quality online platform which is easy use, helps them to organize and share their class notes in one place.

**Functional Requirements:**

1. As a user, I would like to be able to register for a Carnet account.

2. As a user, I would like to be able to login and manage my Carnet account.

3. As a user, I would like my password to be reset if I forget it.

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

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

6. As a user, I would like to be able to easily access Carnet across all of my web-enabled devices.

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

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

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

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

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

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

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

14. As a user, I would like to be able to update pdf images in the notebook.

15. As a user, I would like to be able to delete pdf images in the notebook.

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

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

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

19. As a user, I would like to be able to beautify text over pdf images.

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

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

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

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

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

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

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

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

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

29. As a user, I would like to have a way to connect to the developers for providing feedback

30. As a Software Developer, I would like to have a continuous development and continuous integration pipeline.

31. As a Software Developer, I would like to have a Docker container for local testing.

**Non-Functional Requirements:**

**Architecture and Performance**

We will be incorporating a divide and conquer plan. The whole team is divided into sub-teams: frontend and backend. The backend team will create a RESTful API (Representational State Transfer) written in Node.js. We will be using the express framework and will be incorporating Bluebird for versioning of our API. These frameworks will allow us to create multiples instances of the server in a single deployment. We will be using a fast and scalable database management system called Postgres as it has additional features such as vectors which would allow us to attain faster search results while searching for notebooks and users. We will be following linting rules provided by Airbnb as it would help us to keep our codes organised. We will be incorporating Docker for easy development of our API as it will make the development process independent of the local system. Our goal would be to send a response within 0.002 seconds for each endpoint, when the database tables have at least 100,000 columns and a minimum of 100 users will be sending their request simultaneously.

The frontend will be implemented in React Native, which will be in Typeset. We will be using components for faster loading of pages. These specifications would allow us to export our frontend to Android and iOS apps if the time permits. We will be using one CSS file for all styling as this would allow to organize our files better. We will also be using React Dom and path obscuration methods, which would allow us to increase our security levels.

For both the frontend and the backend we will incorporating a CD/CI pipeline (continuous deployment/ continuous integration). We will be using CircleCI to set up this pipeline. We will be using the hooks provided by CircleCI to connect our gitHub reposistories to it. This would allow us to fully build and test our code after we push our code onto the remote branch. We will try to automate the test cases for more than 95% of the code base. We will be using this as a deployment tool as well. We will give special rights to the master branch so that whenever a commit is pushed to the master branch or a branch is merged to the master branch, after successful execution of all the test cases, the API will be deployed to Heroku while the client will be deployed to gitHub pages.

**Security**

Security is today’s highest point of concern for any quality software. On the backend we will be using bcrypt to encrypt any password using the blowfish algorithm. JWT tokens will be used to authorize and authenticate the user for interacting with the API. We will be keeping a token valid only for five minutes after which the user would have to re-log-in to get a new token. For usability the frontend will manage the token and token renewal till the session is open/Twelve hours have passed. There will be at least two permission levels which will be allowed to interact with the API. A middleware will be implemented which will check for token validation for all routes except the getToken route. For database protection we will be using official modules from npm which will allow us to prevent SQL injection. We will not be using string manipulation to create the SQL statements, We will be using ‘?’ character and will be passing the values to the modules to handle the rest. We will be disabling CORS which will prevent cross-site-scripting and we will be scanning all files to prevent file upload vulnerabilities. We will be taking measures to prevent directory traversals, Denial of service attacks, etc.

In the front end we will be incorporating minified code to make it difficult for people to breach our security barrier. we will be using pseudo paths in the URL to prevent anyone to know the type of page(.html, .php, etc) and other security features will also be used.

**Usability**

Usability is an important part of any software. We will try our best to create a UI which would help any teenager to be fast while also allow anyone above 65+ years of age to use it at their own pace. The UI will be made to allow any user to easily create an account/ work with different notebooks/ search notebooks and share his/her views about a particular notebook. We will make sure that the user is able to create notes on the fly, they are able to write over pdfs in a fast manner. This is one of the main reasons that we will be creating out front-end in react as it is much faster. We have to make the software usable enough that any student and start taking notes for their class on the go and wouldn't have to wait time setting it up every time they want to create some notes. We will also allow the user to easily search any public notebook by its name, owner or any other search parameter.

**Hosting/Deployment**

We will be hosting the Carnet API on the free version of Heroku. we will be deploying two instances of the server, one will be called staging which will be used by our frontend team to test their code. another deployment will be the main which will be open to the public for usage. Heroku also allows for small Postgres database with 10,000 rows. If a higher capacity DB will we required then we will host our DB on AWS. As we will be implementing a CD/CI pipeline, a hook will be created by our CircleCI integration, which will ensure that our master branch is deployed in both the staging and main instance of our server. Environment variables will be used by the frontend to connect to the right deployment.

The frontend will be hosted by Github pages as it will be free and customizable. Again a hook will be created in CircleCI which will ensure that the tested frontend will be hosted on the Github pages.