Presentation Script [draft]

~30s

Hi, My name is Min-Chun Lo and I will be presenting my project on vFlashCards, a web application that brings convenience to students on the go.

The application based flashcards is built using:

1. PostgreSQL
2. Express
3. React
4. NodeJS

As to why? Well, flashcards can help with active recall and having them organized in once place can be incredibly convenient. Not to mention that they won’t be easily misplaced.

~17s

Here is just a view of the original mockup. It takes us from the landing page, sign up, and into the user profile. And a quick view of the library, and function to add card set and individual cards.

~1:24s

Now we will go over the backend.

First the database. A view of the initial database schema and the evolution since inception. The changes include:

* removal of redundant table as per recommendation.
* Recreating database schema due to the switch of diagram tool from MySQL Workbench to pgAdmin, which came with the PostgreSQL installation
* vflashcard table revision as per suggestion to include an ENUM type, which allows different view access
* And even shortening table names, which makes typing SQL queries easier.

So it is always useful to save scripts or work along the way. This can be helpful for troubleshooting purpose. For example, the script generated after my first ERD from pgAdmin showed I unintentionally made the table and column names uppercase, which meant that all names needed to be double quoted during SQL queries.

I also keep a database journal, which is useful when I want to revisit anything that I’ve done or just to reuse old SQL queries.

Finally, here is a view of the tables from SQL shell and another view from pgAdmin.

~48s

Now, the backend routes, which were created using Express & Node.js These are used for login, signup, and to handle requests to retrieve or make changes to virtual cards.

Testing was done with Postman to send API request and receive responses. This helped to flush out bugs and ensure that the backend will be ready to handle requests from the frontend. Here is an example of sign up and login. If successful, a JWT Token is returned.

An example to retrieve library set of cards, and the list of individual cards.

Finally, examples to create and update card sets.

~47s

And now, the frontend implementation, which uses React.

Note the work is done mainly under the src directory. This is just for viewers to get a sense of the file structure.

The screenshot lists private public card related components, alert, spinner, and useAuth for authentication purpose.

In the middle, the pages and routes are listed. The names will make sense when the sitemap and flow chart are shown. The right shows a list of routes under the profile, which is a protected route.

Here is a look at the site map.

And flow chart.

Before we began the demo, just a quick view of what the work environment.

~3:22

So, I will now demo the app hosted on Render. First, I will go ahead and sign up for an account. After signing in, the profile will show the personal library. Since the account is new, the library will be empty.

Let’s go ahead and create a new card set. And we will give it a title and relevant access type. Note that the default is always private. However, the material can also be shared publicly. Let’s go ahead and save as private for now. So next, we will create new cards. Let’s enter Sample question… and sample answer.

Let’s save and add another card. For the being, the page requires a manual refresh to clear the fields. This time, let’s use q2 and a2. Save and let’s return to the card set that has been worked on. And viola! There are 2 cards in our demo set. They are showing the question side of the card. And we can show the answer by tapping flip card button. Cool!

Now, if the card needs revision, we can just edit it. We’ll just type update question and updated answer and save! Notice now we have an updated card.

If we want to delete an individual card, simply click delete. At the time the recording, the page needs to be manually refreshed to see the update. Now, if we go ahead and delete all cards, the set will be empty. Now we also have the option to delete a card set that we no longer want.

So even though the personal library is empty, there is a possibility that others may have shared study cards. Under Public Library, we see a list of demo card sets. Just to distinguish between public and private sets, the access type is shown as public. Individual cards have a green button, just to differentiate between public and private cards.

And upon finishing, we can go ahead and log out. And this will take us back to vFlashCards landing page.

Total: 7:22s

~2:50s

So what are the challenges from this project?

I think includes learning:

* the PERN stack.
* Hash password
* JSONWebtoken, which include implementation and security downsides
* API requests and responses.
* React Router
* Building and deploying the application
* And there is a list of others thing that I did not mention.

So to elaborate on that, it took time to try out and understand the mechanics and get different components to work together.

Database. This ranges from setting up PostgreSQL server, implementing an actual database. Trying out different SQL queries, figuring out appropriate data types. Troubleshoot generated script that did not work out.

Working alone can also be a challenge. There is quite a bit of work that had to be done. And anyone working solo will be responsible for generating ideas, planning, designing, coding, debugging, and writing documentation. You won’t be able to delegate tasks to anyone aside yourself. And Time management is very important. However, this can be a good learning opportunity.

I also had a laptop issue during the project.

* Primary laptop had to be serviced offsite for close to a month.
* Backup laptop crashes frequently and is not exactly reliable. (Despite running on Ubuntu Linux.)
* Had to setup, backup, and configure environments on separate machines.

So that was not ideal. But in a way, I was lucky that it happened shortly after the semester began. Because it would have been very chaotic if it happened later.

Despite working solo, this is not truly a solo project, per se. Assistance and guidance were given at different junctures to push the project to ahead. Without these individuals, the project would not have arrived at its current state. A word of thanks to:

Woodley Gelin for perspective on class and web application projects.

Professor Lapid for spending time to discuss, guide, and provide ideas for the project and deliverables.

Akbar Mirza, my CUNYTechPrep TA, for his technical guidance.

Professor Edgardo Molina, for the patience, generosity of knowledge, and time given to advise and guide my project to completion.

To close the presentation, a final look at the project page. The outstanding tasks are mainly stretch goals. Unfortunately, I was unable to get to them. At the bottom, a list of completed tasks.

<https://github.com/users/not-x/projects/1>

(Pause and switch tab!!!)

https://github.com/not-x/vflashcards

Finally, a view of the Github repository. This consists of the documents, notes, application code, and time log.

Thank you for listening!