**Draft**

1. Cover
2. What & Why?
   1. Full stack web application flashcards. (Based on PostgreSQL, Express, React, Node.js.)
   2. Because everything is organized in one place and not easily misplaced.
   3. Convenience of study aids for students on the go.

**UI / Interface / Mockup (no prototype)**

1. Login > Sign-up
2. User profile > View Private lib
3. View sample set of cards > Edit?
   1. Note, public view should not allow edit access
4. Add new set > Create New Card 1 > Create new card 2
   1. Note: Prototype to come later…

**Behind the Scenes / Backend**

Database  
- We see the cards, but where and how is the data stored?

1. Database design (using different tools)
   1. 09-09 - ERD (Initial)
      1. Using MySQL ERD tool
   2. 09-18 – ERD
      1. Deleted the redundant table, as per recommendation from Edgardo Molina.
   3. 10-01 – Recreated using ERD from pgAdmin4 (PostGreSQL). Used an automated script to generate tables. Noticed that double quotes were needed for all SQL queries. Deleted existing tables and began recreating them.  
        
      Revision:   
      Recreated using ERD tool from pgAdmin4 (PostgreSQL), which can generate script to create tables. Noticed that all tables required double quotes for all SQL queries, which was a hassle. Deleted all existing tables and recreated them scratch.
2. Database design (cont'd) - Using ERD Tool from PostGreSQL (via pgAdmin 4)
   1. 10-09 - Changed composite primary key to a using a single attribute as a workaround for failure to create the 3rd table, vflashcard.
   2. 10-17 – Added additional attribute under vflashcard\_set for view access. Implemented an ENUM type (public/private) as suggested by Edgardo Molina.
   3. 10-29 - Shorten table names for convenience. (Less typing during coding & debugging processes.)
3. PostGresSQL Database - vFlashcards (screenshots)
   1. Database tables
   2. Show queries.
   3. Other work that took place (may require a new slide for this)
4. Routes - Login & Authentication (using Express)
   1. Login.js - Code snippets
      1. User sign up
   2. tokenGenerator - Code snippets
      1. For authentication and integrity
   3. auth.js - Code snippets
      1. Where verification takes place

**Routes in Action (using Postman)**

1. Sign up > Login > Retrieving a card set from the library
2. Viewing a card > Adding to an existing card > Updating a card
3. Delete a card

**Challenges**

1. Challenges

Learning / testing components

* + 1. JSONWebToken – Took > 10 hours to learn, troubleshoot, and understand the mechanic.
    2. Database – Had to augment existing knowledge (from college course) with additional tutorials. These included PostGreSQL server setup, creating tables, using the appropriate data type, etc.
  1. Working solo - responsible for all aspects of the project including ideation, planning, designing, coding, debugging, and documentation. Tasks can be incredibly time consuming.
  2. No teammates.
     1. Lack of sounding board, which can be very useful when one is stuck.
     2. Responsible for all follow ups with TA and supervisor.
     3. Useful to have other opinions and perspectives.

**Wrap up**

1. Anticipated vs current roadmap
2. Task log
3. Project management board
4. Actionable items
   1. Frontend
   2. Tentative - Strech goals