RICE EXPLOSION: Nakib Abedin, Wilson Mach, Donald Bi
SoftDev
P00: Half-Quick
Time Spent: 2.5 hrs
2022-11-02

Target Ship Date: {2022-11-11}
Scenario 1

Database:
users:

- Contains two fields, one for usernames and one for passwords

stories:

- One field for the id (makes sure each story has a unique identifier)
- One field for the username of the person who created the story
    - Taken from the *users* table
- One field for the story text (a list of containing each parts of the story)
- One field for the contributors (a list with contributors whose index corresponds to their story text)

Python Files: (file names are for reference only)
app.py: file designed to run the flask server and send information to the front end

- Will also include login session code (where a logged in user could see and do specific things a non-logged in user can't)
- Will perform the log in form in login.html

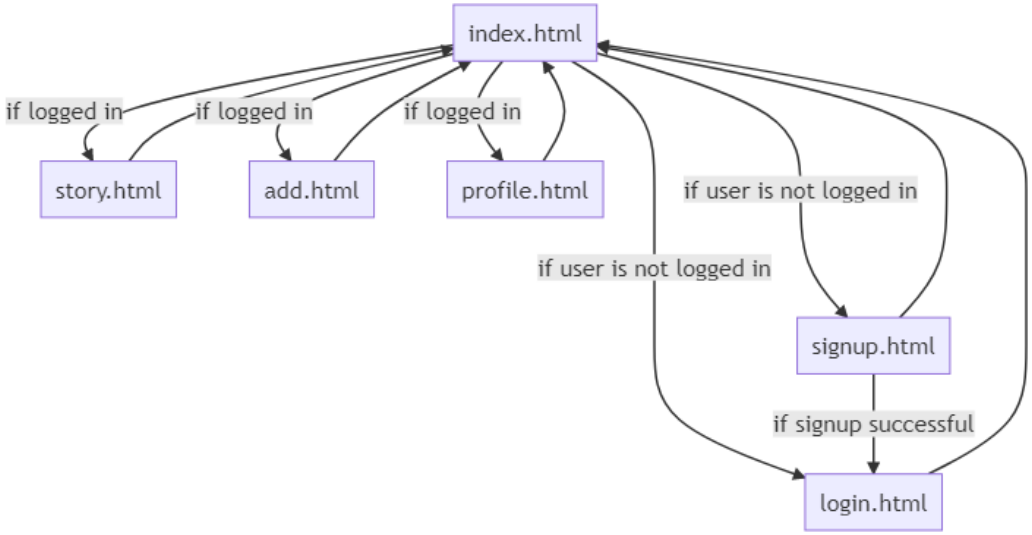
db.py: file with functions to retrieve information from the database

- Will format data from the *stories* table into 'cards' (box-like display) that will be shown on the front-end

edit.py: file to edit the information in the database

- When logged-in users try to add stories then this file will perform the task

Site Map:



Front End:

index.html

- Displays a big welcome in the center when you first enter, prompting you to log in
- Login <button> at the top right (if you are not logged in) that will bring you to *login.html*
- Sign Up <button> (if you are not logged in) that will bring you to *signup.html*
- Random story titles that other users have created appear if logged in, the information will be taken from *db.py*
    - Clicking a story will send the story's id to *story.html*, where it displays the story's information (title, story, creator)
- Add story <button> at the top left that sends you to *add.html* and *login.html* if not

Signup.html

- Contains a <form> where you can make a new account
- Button to submit the form, checking to see if the user already exists and adding it to the database if it doesn't

login.html

- Contains a <form> where you can input your username and password, the password will appear as aster****!!
- Button to submit the form, and will connect to *db.py* to retrieve user info
    - If the user info is correct, log them in and return to *index.html*
    - If the user info is not correct, display a mean message and keep them logged out

story.html

- Contains a template where a story is displayed

- Takes information from *stories* table in the database using the id that it receives from *index.html*

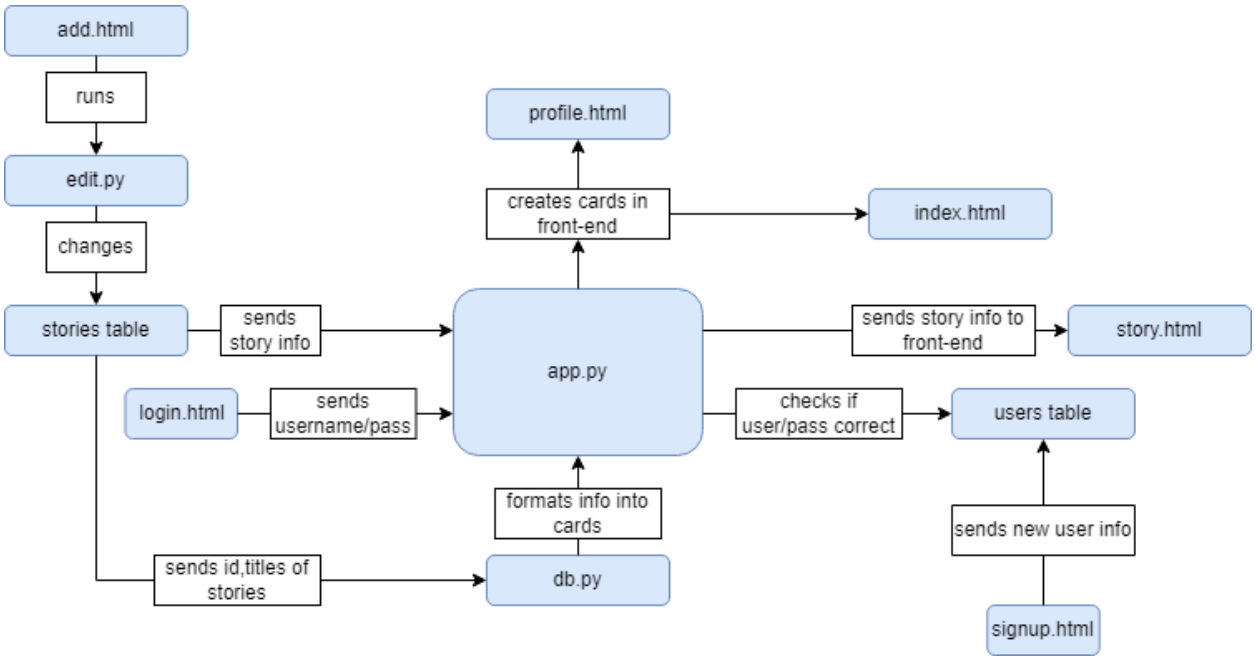
  <u>If have not contributed</u>
  - Shows only the latest addition
  - <form> to input an addition to the story, appending it to the list in *stories* table

  <u>If have contributed</u>
  - Shows the whole story and text saying you have already contributed

add.html
- Contains a <form> where you can input title and the beginning of the story
- Button to submit the form and create a new story! (adds it to *stories* table)

profile.html
- Shows all the stories you've created
- Shows all the stories you've contributed to

Concept Map:



To do List:

1) Front End: (Nakib)
    a) index.html
        i) Displays list of stories to user
    b) login.html
        i) Has a form to log in
    c) story.html
        i) Displays story to user
    d) add.html
        i) Page that runs add.py to update stories database
    e) signup.html
        i) Has a form to create a new user
    f) profile.html
        i) Shows the stories that the user created and contributed to
2) Back End:
    a) Story Database (Wilson)
        i) Takes entries from the csv file from k18
        ii) Takes in new data from user and adds to story database
    b) User Database
        i) Takes in new user data in user database
        ii) Checks user data in user database
    c) Python (Donald)
        i) app.py
            (1) Sends info to front end and log in
        ii) edit.py
            (1) Updates stories database
        iii) db.py
            (1) Uses databases to perform things(format story cards)