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

Target Ship Date: {2022-15-11}
Scenario 1

Database:
users:
- Contains two fields, one for usernames and one for passwords

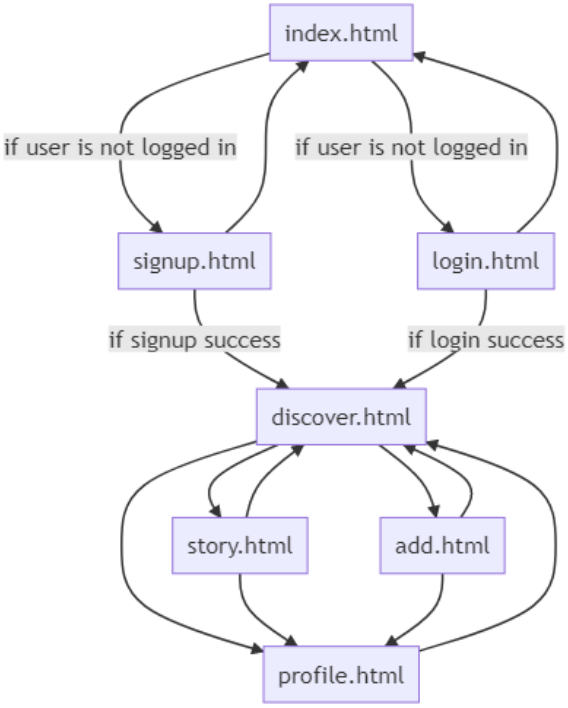
stories:
- One field for the id (makes sure each story has a unique identifier)
- One field for the title
- One field for the creator's username

Story_info:
- One field for the id
- One field for the story's part (which contribution it is on)
- One field for the part's text
- One field for the part's contributor

Python Files: (file names are for reference only)
app.py: file designed to run the flask server and send information to the front end
- Includes the routes and uses functions from *db.py* to perform things like logging in or formatting story cards (based on the sitemap)

db.py: file with functions to retrieve/send/use information from the database
- Has functions to check database for credentials in login/signup
- Has functions to create/add to stories
- Has functions to format data from the *stories* table into 'cards' (box-like display) that will be shown on the front-end
- Has get functions to send info to *app.py* to send to frontend

Site Map:



Front End:

index.html

- Displays a welcome message
- Login <button> (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*

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; sends you to *discover.html* if signup was successful

login.html

- Contains a <form> where you can input your username and password, and 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 go to discover.*html*
    - If the user info is not correct, display message and keep them logged out

discover.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*
- Create story <button> at the top left that sends you to *add.html*

story.html

- Contains a template displaying the story's information (title, story, creator)
- Takes information from the *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, adding new entry to *story_info* table

  <u>If have contributed</u>
  - Shows the whole story

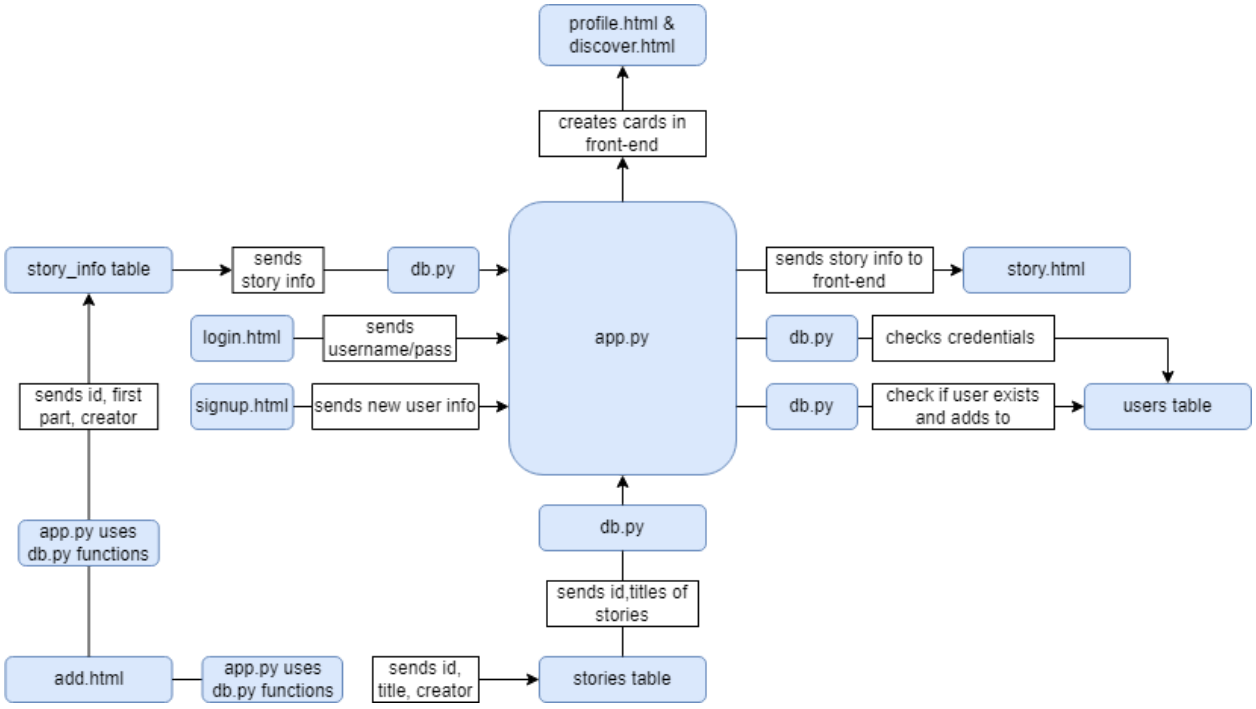
add.html

- Contains a <form> where you can input the title and the beginning of the story
- Button to submit the form and create a new story! (adds it to *stories* table an also creates the first entry of the story in *story_info*

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) db.py
            (1) Uses databases to perform things(format story cards)