Bug Busters: Victor Casado, Ethan Sie, Marco Quintero, Qianjun Zhou
SoftDev
P00: Move Slowly and Fix Things
2024-10-28
TARGET SHIP DATE: 2024-11-11

Program Components

Frontend Components:

1. HTML
   a. homepage.html — Displays a list of stories and provides navigation options.
   b. login.html — Allows users to log into their account and contains a button to go to signup.
   c. signup.html — Enables new users to create an account.
   d. viewStory.html — Shows a story in its entirety, including the title and main text.
   e. createStory.html — Provides a form for users to create a new story.
   f. editStory.html — Allows users to edit a story once.

Backend Components:

1. Flask/Python - Stored session with username, password, and if we just signed in
   a. "/" (Root)
      i. If no session found, redirect to login
      ii. If session found, if we just did login, if user/password combo is correct, do not redirect
      iii. If session found, if we just did login, if user/password combo is incorrect, redirect to login
      iv. If session found, if we just did sign up, if user/password combo is unique, add user/password to db
      v. If session found, if we just did sign up, if user/password combo is not unique, redirect to login
   b. The remaining are described through their names

2. SQLite DBs
   a. User Table: Userid (unique), username (unique), password
   b. Stories Table: story id (unique), title, main text, most recent entry, creator (?)
   c. Stories Contributed To: Userid (unique), edited story1? (boolean), edited story2? (boolean), … (each column represents one story)

Unique - we will implement a check before adding to make sure there isn't one that's the same.
Boolean - stores a True/False instead of a string, or stores something that is easily able to be turned into a True/False
Nothing - Simple String

Site and Component Maps (Used: lucid.app):

HTML Webpages	HTML pages
-takes input from the Flask to make a webpage -CSS to make the webpage look nice -takes in inputs to send back	homepage.html login.html signup.html viewStory.html createStory.html editStory.html

Python/Flask	App Routes
-renders the HTML files with appropriate inputs -recieves data from webpage -keeps track of cookies/sessions -modifies the database to keep it updated -redirects pages to each other -checks inputs to comfirm validity	"/" "/login" "/signup" "/viewDoc" "/editDoc" "/createDoc" "/logout"

Database
userID(int) + username(String) + password(String) storyID(int) + title(String) + mainText(String) + recentEntry(String) userID(int) + editedStory1(boolean) + editedStory2(boolean) + ...

Tasks:

HTML/CSS (makes website look cool and displays all vital information): Victor

Flask/Python session-based logic (allows you to log in and stay logged in): Ryan

Flask/Python other logic (connects HTML code with the SQL code so website has information): Ethan

SQLite (handels storing of data and creates simple data manipulation methods): Marco

Top-left page

Site and Component Maps:

Diagram boxes: edit → homepage → Signup → Login; homepage → Create, View

Handwritten (blue): GREAT DIAGRAMS!!
Handwritten (red): Very nice visuals!
Handwritten: Simple + Easy to Understand
Handwritten (red): Q – What diagram tool used?

- HTML Webpages — takes input from the Flask to make a webpage / CSS to make the webpage look nice / takes in inputs to send back
- HTML pages — homepage.html, login.html, signup.html, viewStory.html, createStory.html, editStory.html
- Python/Flask — renders the HTML files with appropriate inputs / recieves data from webpage / keeps track of cookies/sessions / modifies the database to keep it updated / redirects pages to each other / checks inputs to confirm validity
- App Routes — "/", "login", "signup", "viewDoc", "editDoc", "createDoc", "logout"
- Database — user(ID#) + username(String) + password(String); story(ID#) + title(String) + mainText(String) + recentEntry(String); user(ID#) + editedStory1(boolean) + editedStory2(boolean) + ...

Tasks:
HTML: Victor
Flask/Python session-based logic: Ryan
Flask/Python other logic: Ethan
SQLite: Marco

Top-right page

Handwritten (blue, top right): Reviewed by: Indigo

Bug Busters: Victor Casado, Ethan Sie, Marco Quintero, Qianjun Zhou
SoftDev
POO: Move Slowly and Fix Things
2024-10-24
TARGET SHIP DATE: 2024-11-11 *(handwritten: Cred. Date!!)*

Program Components

Frontend Components:
1. HTML
   a. homepage.html
   b. login.html
   c. signup.html *(handwritten: Can ✓)*
   d. viewStory.html
   e. createStory.html
   f. editStory.html

Backend Components:
1. Flask/Python
   a. "?"
      i. Redirects to login if no session found *(handwritten: can they view the page if they're not logged in?)*
      ii. If a session is found, but the user/password combo is incorrect and we didn't just sign up, redirects to login
      iii. If a session is found, but the user/password combo is incorrect, and the user already exists, redirect to signup *(handwritten: should be log in instead? Maybe)*
      iv. If a session is found, but the user/password combo is incorrect and we did just sign up, adds user/password to db
      v. If a session is found with a correct user/password combo, do not redirect
      vi. Sends the status (has the story been edited / can you edit) of stories to frontend
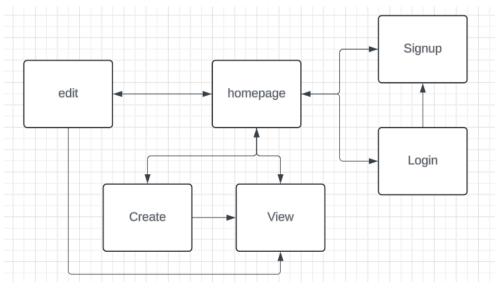   b. The remaining are described through their names

2. SQLite DBs
   a. User Table: Userid (unique), username (unique), password
   b. Stories Table: story id (unique), title, main text, most recent entry, creator (?)
   c. Stories Contributed To: Userid (unique), edited story1? (boolean), edited story2? (boolean), ... (each column represents one story)
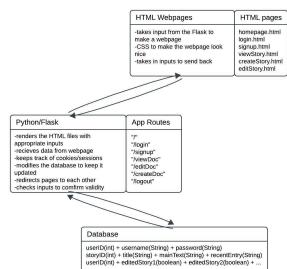

Bottom-left page

Site and Component Maps:

Diagram boxes: edit → homepage → Signup → Login; homepage → Create, View

Handwritten (red): redirects to homepage?
Handwritten: which is for? send redirects
Handwritten: send back what?

- HTML Webpages — takes input from the Flask to make a webpage / CSS to make the webpage look nice / takes in inputs to send back
- HTML pages — homepage.html, login.html, signup.html, viewStory.html, createStory.html, editStory.html
- Python/Flask — renders the HTML files with appropriate inputs / recieves data from webpage / keeps track of cookies/sessions / modifies the database to keep it updated / redirects pages to each other / checks inputs to confirm validity
- App Routes — "/", "login", "signup", "viewDoc", "editDoc", "createDoc", "logout"
- Database — user(ID#) + username(String) + password(String); story(ID#) + title(String) + mainText(String) + recentEntry(String); user(ID#) + editedStory1(boolean) + editedStory2(boolean) + ...

Tasks:
HTML: Victor
Flask/Python session-based logic: Ryan
Flask/Python other logic: Ethan
SQLite: Marco

Handwritten (red): What is this? Is this another table? What does the boolean represent? How will you use this info?

Bottom-right page

Handwritten (red, top): REV'd by Boolean Boas

Bug Busters: Victor Casado, Ethan Sie, Marco Quintero, Qianjun Zhou
SoftDev
POO: Move Slowly and Fix Things
2024-10-24
TARGET SHIP DATE: 2024-11-11

Program Components

Frontend Components:
1. HTML
   a. homepage.html
   b. login.html
   c. signup.html
   d. viewStory.html
   e. createStory.html
   f. editStory.html

Backend Components:
1. Flask/Python
   a. "?" *(handwritten: INCONSISTENT (?))*
      i. Redirects to login if no session found
      ii. If a session is found, but the user/password combo is incorrect and we didn't just sign up, redirects to login
      iii. If a session is found, but the user/password combo is incorrect, and the user already exists, redirect to signup
      iv. If a session is found, but the user/password combo is incorrect and we did just sign up, adds user/password to db
      v. If a session is found with a correct user/password combo, do not redirect
      vi. Sends the status (has the story been edited / can you edit) of stories to frontend
   b. The remaining are described through their names

2. SQLite DBs
   a. User Table: Userid (unique), username (unique), password
   b. Stories Table: story id (unique), title, main text, most recent entry, creator (?)
   c. Stories Contributed To: Userid (unique), edited story1? (boolean), edited story2? (boolean), ... (each column represents one story)