

Team Green Monkeys; Daniel He, Faiyaz Rafee
SoftDev
P00-SCENARIO ONE
2022-11-2
time spent: 1.5 hrs

Target ship date: {2022-11-18}

Components of SCENARIO ONE:

_Users and login functionality using SQLite

_Displaying stories, adding on/ starting stories with the following features:

- Full stories can only be read after contributions
- Only last contribution displayed when adding on to a story
- Categorized into a predetermined set of genres (selected from a predetermined dropdown list using html select tag), alphabetical, etc
- Word limit when adding on, not applicable when creating a story
- Only contribute to a story once
- Must be logged in to access stories
- Title, body, authors displayed, date and time

_Homepage: body displays all stories (display in a card form, clickable to view in its entirety)

_Extra features: likes, comments, how many visited, how many contributed, how many contributed after you contributed, profile icons. “how many...” features to be implemented by adding a counter that increases each time a method is run. “How many contributed” features would necessitate an extra column to be added to certain tables linked to stories.

Software COMPONENTS to achieve:

_Flask web server serving different routes/ webpages

_Databases storing all stories and affiliated resources, users

_Python handling:

- Editing SQLite databases, (updating existing story databases)

_Links acting as buttons for navigation and activating methods

- Html and CSS for UI

_Forms for story writing (error handling for unfilled forms)

_Templates using jinja to insert dynamic, changing data into stagnant html, e.g. user contributed stories, available stories, etc

FLASK; work allotment: BOTH [flask], DANIEL [database], FAIYAZ [CSS]

App.py

Rudimentary **ROUTES** logics

- Homepage (Faiyaz):

- Accesses argument: username
 - Access table Users to specify selection in Contributed_Stories table
 - Display all contributed stories in HomePage Template
- Stories (Faiyaz):
 - Accesses argument: genres, alphabetical, or none—to specify returned data
 - Accesses database table of all stories' title and table Stories, but making sure to avoid titles already in Contributed_Stories table
 - Display data in Stories Template
- Adder_logic (adding logic) [Daniel]:
 - Accesses arguments: username, story title, date and time
 - Updates corresponding Stories table
- Creator_logic (creating logic) [Daniel]:
 - Accesses arguments: username, date, genre, title of story, story body
 - Appends data to corresponding table: ContributedStories, Stories, Table of all story titles
- Login
 - Accesses arguments: username, password and somehow sends that to the rest of the python
- Signup
 - Accesses arguments: username, password and appends data to Users table

Templates:

HomePage
 Story_Maker
 Story_Adder
 Stories

Update_data.py

Give_me_data.py

DATABASE Tables:

“Users” (2 columns):

- Username TEXT P.K. | password TEXT not null

“<User>'s_Contributed_Stories” (a table unique to each user) [1 column]:

- Title of contributed stories for corresponding user TEXT P.K.
 - Currently separated from “Users” table so appendage of data can be facilitated through adding new rows instead of changing data in existing columns

“<Story Title>”, each story is linked to a unique table (4 columns):

- Author username TEXT P.K , date TEXT not null, body of TEXT not null, genre TEXT not null

“All_Stories”, table of all story titles (for sorting into categories) [2 columns]:

- Title of story TEXT P.K, genre TEXT not null
 - Currently separated from “<Story Title>” table so appendage of data can be facilitated through adding new rows instead of changing data in existing columns when a story is updated

RELATIONSHIPPING:

_App.py will handle all the logics of navigation, as well as logics of actions, such as editing or updating forms.

_Updating or editing forms will call upon methods of Update_data.py

_When navigated to the correct routes, articles will be displayed by fetching data in the database through methods of Give_me_data.py



