xenus: Biraj Chowdury, Nahi Khan, Henry Liu, Albert Wan

P#02 - The End
SoftDev1 Pd9
2020-1-1

Escape The Room

Overview:

The overall goal of the project is to create a fun game with a storyline where players attempt to escape the universe Xenus. We will have puzzles like type racer, Galactica, trivia, chance, blackjack, snake and computation problems which will be timed and will only be available once the player completes the puzzles before that one. We will be using *Bootstrap* as our front-end framework because our best front end developers are most familiar with this framework.

APIs Overview:

1. Diceful API
    a. Will be used for our chance game
    b. This API provides dice rolling functionality.
    c. You can simulate rolling 6-sided or 20-sided die.
    d. The API will return a JSON object with the results of the roll.
    e. No keys, no quotas
2. Open Trivia API
    a. Will be used for our trivia game
    b. Quota: 50 questions in one call
    c. Entry includes: category, difficulty, type, question, answer (correct & wrong)
3. NBA Player API
    a. Will be used to create profile pictures
    b. No quota, no keys
    c. Provides headshots and basic information of NBA players
    d. Will be used for profile pictures
4. Deck of Cards API
    a. Will be used for our card games
    b. No key

Player Experience:
Users would need to login or signup to be able to play the game. Once the session begins, The user can see what level they are on and can play games to move up levels.

Once they have completed all levels, they have successfully escaped Xenus. Users can also change their profiles on their profile page.

In Depth View of Games:

1. Type racer
    a. Player will type up sentences given by the computer
2. Galactica
    a. Player will attempt to survive as long as possible
3. Trivia
    a. Open Trivia API
    b. Player will solve multiple choice problems
4. Chance
    a. Diceful API
    b. Players will roll a die to test their luck
5. Blackjack
    a. Deck of Cards API
    b. Players will need to play Blackjack to make money
6. Snake
    a. Source: https://gist.github.com/straker/ff00b4b49669ad3dec890306d348adc4
    b. Players need to play snake in order to collect resources
7. Computation problems
    a. Basic arithmetic problems in a multiple choice like test

Roles:

1. Henry:
    a. Project Manager
        i. Revise design doc
        ii. Assign tasks
        iii. Facilitate communication
    b. Troubleshoot issues and complete minor coding tasks as necessary
2. Nahi:
    a. Front-end
        i. Templates using Bootstrap
            1. Make a template that will store and format the title, headers, images, and forms to be displayed on a given page■ Renders the template for each page requested
        ii. Create cards for each API used to store in the Knowledge Base
    b. Game Designer
3. Biraj:
    a. Back-end

i. Database Operations Module (Insert to Table, Edit a Row, Create a Table)

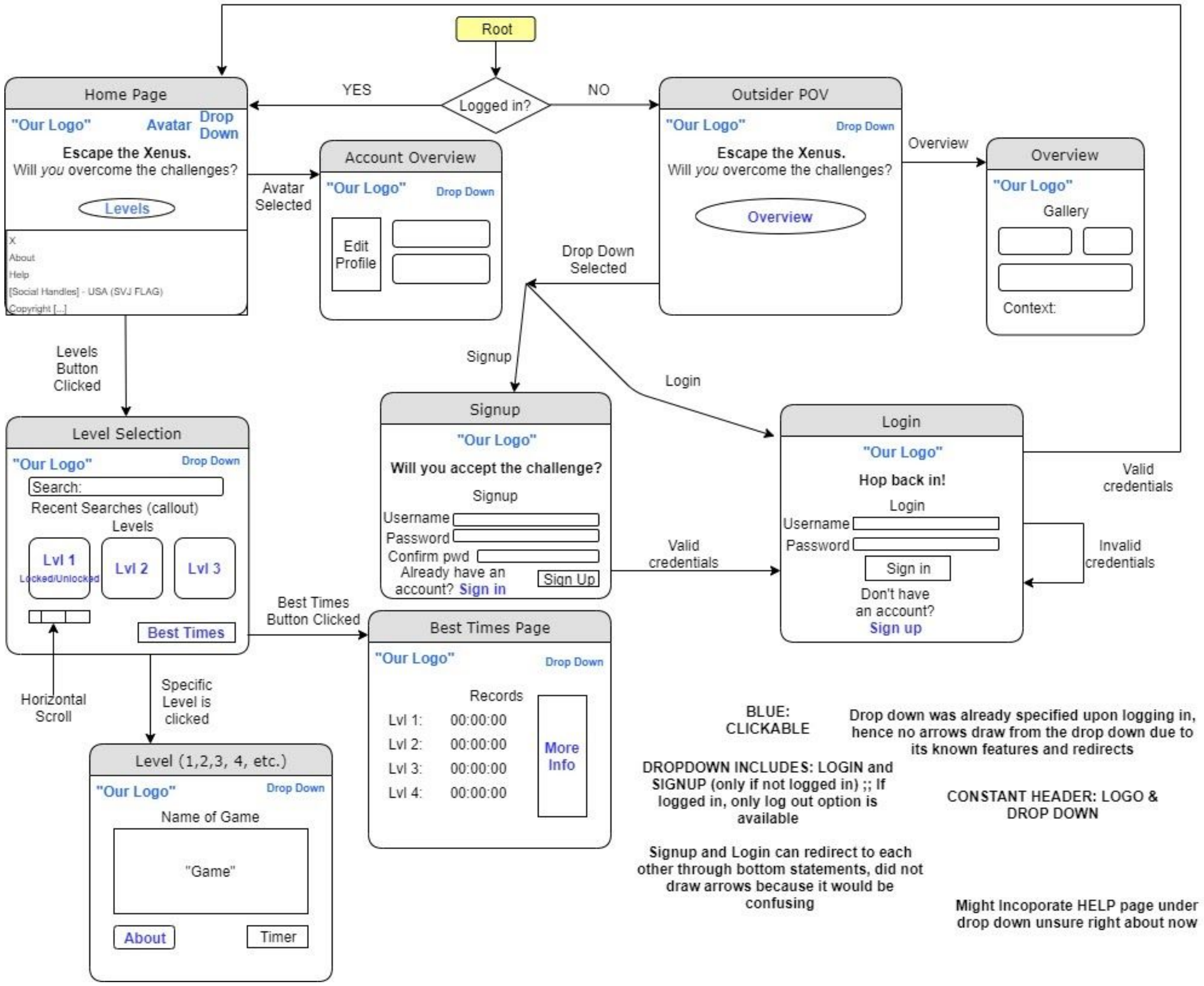
ii. Facilitate any changes that the user will request, such as adding a new row in a database when the user pulls information from the API to store that information and facilitate faster access in the future

iii. Open the url for each API every time a call is made and pull and store needed information from the API in the appropriate database

b. Game Designer

4. Albert:
   a. Assist Nahi with front-end
      i. Work on presentation and aesthetics
   b. APIs
      i. Obtain access to keys for all APIs
      ii. Pull relevant information from APIs
      iii. Add API cards to Google Drive
   c. Game Designer

Site Map:



BLUE: CLICKABLE

Drop down was already specified upon logging in, hence no arrows draw from the drop down due to its known features and redirects

DROPDOWN INCLUDES: LOGIN and SIGNUP (only if not logged in) ;; If logged in, only log out option is available

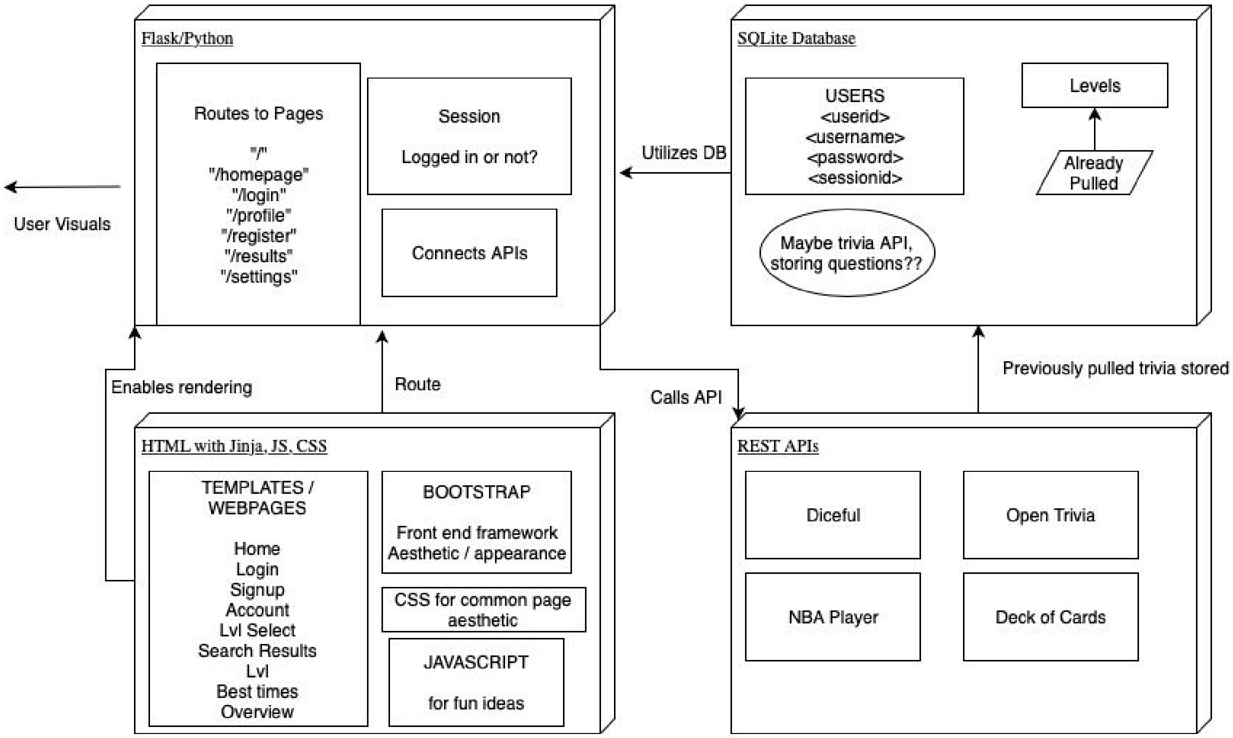
CONSTANT HEADER: LOGO & DROP DOWN

Signup and Login can redirect to each other through bottom statements, did not draw arrows because it would be confusing

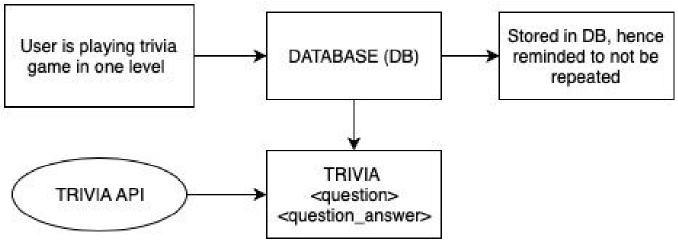
Might Incoporate HELP page under drop down unsure right about now

This is a massive site map, I know I have to display the search results page so please go easy on me Also, the map is crammed only because of fitting sorry :c

Component Map:



EXAMPLE:



User is playing trivia game in one level
DATABASE (DB)
Stored in DB, hence reminded to not be repeated
TRIVIA API
TRIVIA
<question>
<question_answer>

Each of these components will be dealt with by a different member of the team, so that in the end they can be pulled together to form the entire website. Communication will be required between SQLite and the API, as well as the HTML and Flask code in order to render templates correctly.

Database Diagram:

USER

username primary_key	password

“bertw2002”	“pass123”
“coolusername”	“password123”
“Xenus”	“passsss”

STORIES

storyid primary_key	title	text
0	A day in the life	Enter story
1	Funny jokes	Enter story
2	How we coded this project	Enter story

QUESTIONS

QUESTIONID primary_key	question	a	b	c	d	correct
0	What is 9 * 3	27	14	12	14	a
1	When did Steve jobs die	1900	1922	2000	2011	d
2	How many minutes in a day	36000	1440	14400	36000	b

The **Primary Key** for each table shows that each entry is required to have this item; it cannot be null or empty.