Tnpg: Team JBEE
Roster: Endrit Idrizi, Vedant Kothari, Benjamin Rudinski, & Ziyad Hamed
TARGET SHIP DATE: {2024-12-18}

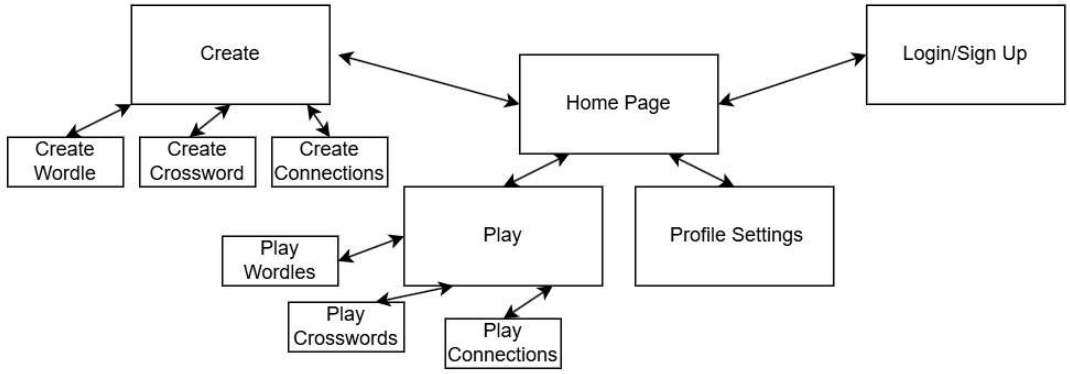
Description:
Our project is essentially a more advanced version of the New York Times 'Games' section. It will contain games such as *Wordle, Mini Crossword,* and *Connections*. Each of these games will have the same rules and instructions as those found on the official NYT website, but in our website, the kick is that any user can create and customize their own games. In fact, they can publish it to the site to make it accessible for other users to play as well, providing that they are logged in properly. Regarding the APIs, we will be using Merriam Webster Dictionary API. It will be used to check for correctness in words. By logging in, users will have the ability to create their own games as well as view their completed games from the past. We might even implement a feature with credits, where users can earn credits based on the games they play, difficulty, and time played, but this feature will not be possible if you aren't logged in. In addition, in times of happiness, the following songs could be played as a sign of congratulations:

- Protagonist - Blanco
- Evicted - Nemmzz
- 4am - JBEE
- SCAR - Song Bird, Gyakie & JBEE
- Talking Stage - JBEE
- Just 4 Me - JBEE
- Changed - JBEE & CRYSTAL MILLZ
- Next Up - S4-E2, Pt. 2 - JBEE
- Heart On Ice - JBEE
- With Me Or Not? - JBEE

Our music will be played using MP3 files and HTMLs audio tags.

APIs of Choice:
Merriam Webster Dictionary API - This will be used to check that real words are being used in the making of wordles, connections, and crosswords.

Site Map:



Component Map:

[Frontend (HTML Templates + CSS)]
↓
[Javascript Logic] <- User input validation -> [Middleware]
↓
[Flask Backend] <-> [APIs]
↓
[SQLite Database] <- stores -> User Data, Game Data, Custom Games, Music

Interactions

- **Frontend ↔ Middleware:** Captures user input and sends HTTP requests to the middleware
- **Middleware ↔ Backend:** Processes data from the frontend, sends API calls, and retrieves database records
- **Backend ↔ APIs:** Handles API calls for word validation, data fetching, and visualizations
- **Backend ↔ Database:** CRUD operations for storing and retrieving user, game, and music data

Front-End Framework:

We will be utilizing Foundation for this project because during k23 we realized that Foundation is easy to implement yet contains a lot of customization and flexibility. If you look at NYT games specifically, there isn't much in terms of the CSS which pops out, which is why we find it okay to go with Foundation for this project. We also think it may be better to use Foundation for the project if we want to make it look unique, due to the endless possibilities of things that we can do

with elements such as the navbar and other scrolling features of the website.

Database Organization:
Role: Storage for user accounts, game information, and progress.

- Database will keep track of user accounts and passwords, and different user made and pre populated games.
- Game information and parameters, what games users have played.
- User statistics

SQLite3 Tables:

Main Components	**Example**
Users [via ID]	[UserID1, UserID2, UserID3]
Games e.g. Connections [via ID]	[GameID1, GameID2, …]

The table above represents the main components within the website as it stores all of the users and games that have been uploaded or pre populated.

User	**Example**
Games Created [via ID]	Table(UserConnectionsID4)
Games Completed [via ID]	Table(UserConnectionsID1, UserWordleID2)
User Stats e.g. Wordles Won on First Attempt	2
Password	********
Username	“JBEE”
Login Timestamps (in Unix Epoch)	Table(145624542, …)

The User table above stores information on the user, such as the amount of games that user created, past games completed(played), user stats, password, and username. This information will be used to facilitate gameplay as well as personalize user

experience. Login timestamps will be measured in unix epoch to measure user logins and contribute to user stats.

User Games	**Example**
Title	“My first Connection”
Pairs	“red, blue, yellow, orange; cat, dog, turtle, hamster”
Author	“JBEE”
Difficulty	“Hard”
Type	“Connection”

The ‘User Games’ table above is responsible for storing the user’s created games to be managed and shared with others. It takes parameters like ‘Type’ and ‘Difficulty’ to provide context for the application to properly handle it.

Task Assignments:

TASK	Vedant Kothari	Endrit Idrizi	Benjamin Rudinski	Ziyad Hamed
Set up Flask and SQLite3 environment			X	
Build User Authentication Functionality				X
API Configuration and Connections		X		
Middleware file organization				X
Java Script		X		
Build Database				X
Frontend (HTML Templates)	X			
Frontend (CSS + FEF)	X			
Interconnections (making sure everything actually connects and works together)		X		