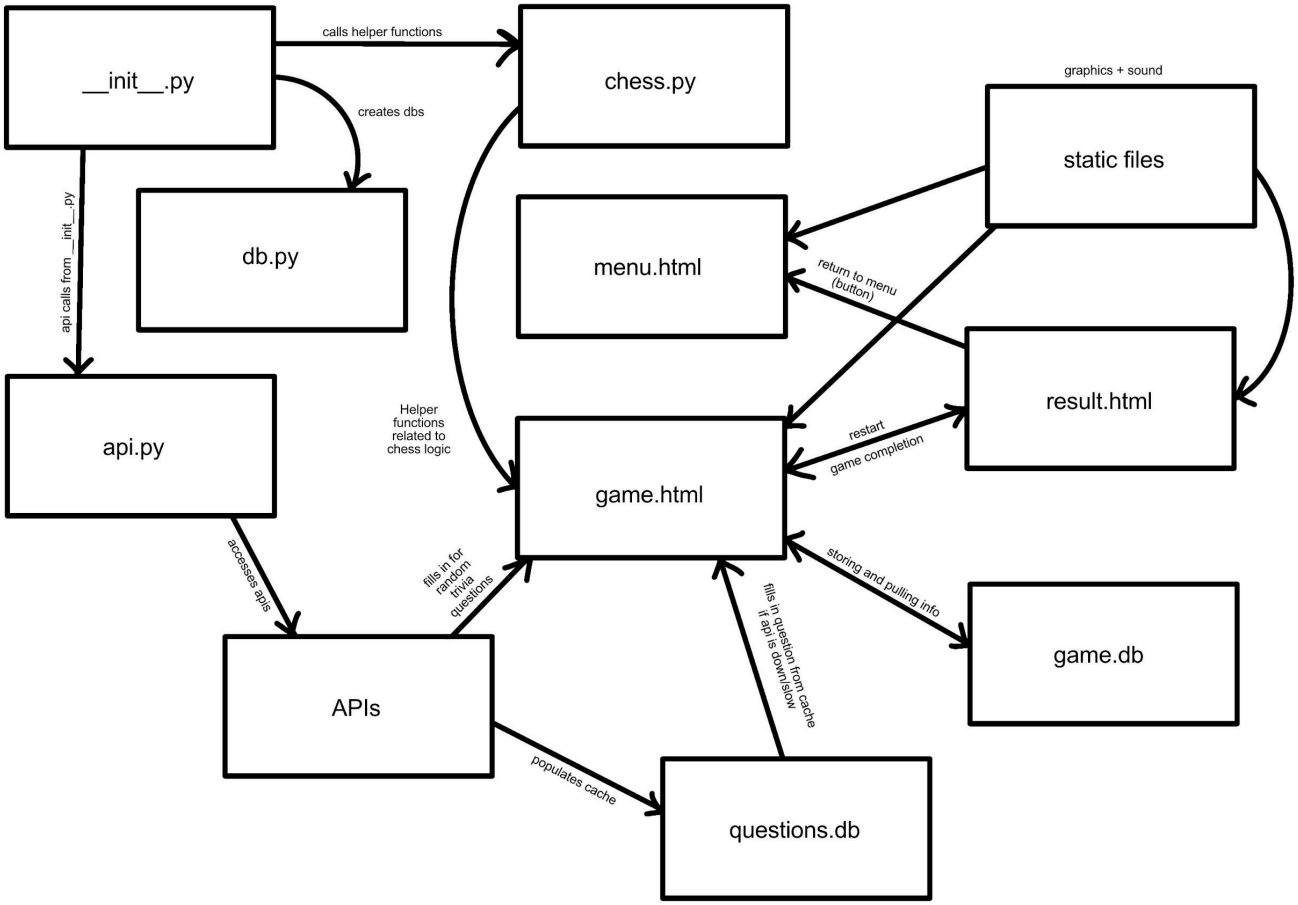
Chrivia by Peaked in Elementary School (PIES)
TARGET SHIP DATE: 2025-12-22

Idea for site
The user will be able to play chess locally, but every time they make a move, they have to answer a trivia question. If they fail the multiple choice trivia question or take too long, they will not be able to move.

Program Components
- *HTML Pages* (contain css, js, and any FEF)
    - menu.html (The page where the player goes to first and contains the play button)
        - Will be the main method of navigation (access different gamemodes, types, etc.)
        - Effectively acts as our landing page
        - A play button will be displayed allowing the user to begin the chess game with the selected settings (selected settings are an optional addition)
        - Will make usage of static files for both graphics and audio
    - game.html (where the game is played)
        - An intractable page that will allow the user to navigate a game of chess (between two people, unfortunately without online connection) hopefully without having to refresh the page
        - Will primarily be making use of JS and CSS to display animations (such as spinning the board and moving the pieces) as well as sending information back (upon different interactions) to __init__.py
        - Will contain modals which allows display of trivia questions and corresponding responses
    - result.html (where the result of the game is displayed)
        - Maybe contains a snapshot of the game
        - Will simply display the result (player 1 wins!, player 2 wins!) as well as contain a button that allows the user to either return to menu or restart the game directly
        - Might be removed entirely
- *DBs* (contains the all relevant information and supports the running/management of the game) – Explanations can be found in the DB section of our DD <3
    - game
    - questions
- *APIs* (provides all information for the trivia; we plan to extract data from each of the APIs to create random questions that follow our specific question formats) – In the case that any API fails to run, a question will be pulled from the Questions DB
    - MoviesAPI

    - MerriamWebsterAPI
    - CountriesAPI
    - RickAndMortyAPI
    - ChessAPI (if you want to use an engine for best moves)
- *Python Files* (runs the program)
    - __init__.py
        - Runs flask and pages
        - Calls other python files and utilizes them
        - Included helper functions:
            - Fetch random question
            - Fetch page content for display
    - db.py
        - Creates databases
    - api.py
        - API calls
        - Will be called by __init__.py in trivia events
    - chess.py
        - Chess logic (+ helper functions)
        - Will be called by __init__.py when running the game
        - Included helper functions:
            - End game
            - Flip board
            - Reset board
            - Move piece
            - Highlight legal squares
            - Check for checks/checkmates
- *Static Folder* (all supporting files such as images and gifs)
    - board.png
    - white_pawn.png
    - white_knight.png
    - white_bishop.png
    - white_rook.png
    - white_queen.png
    - white_king.png
    - black_pawn.png
    - black_knight.png
    - black_bishop.png
    - black_rook.png
    - black_queen.png
    - black_king.png

Component Map



Database Organization

- Game
    - Stored data on game history (what moves were played)
- Questions
    - Cache for trivia questions

films

INTEGER	id	PK NOT NULL
TEXT	title	NOT NULL
TEXT	genre	NOT NULL
TEXT	plot	NOT NULL
TEXT	director	NOT NULL
TEXT	rating	NOT NULL
TEXT	img	NOT NULL
TEXT	released	NOT NULL

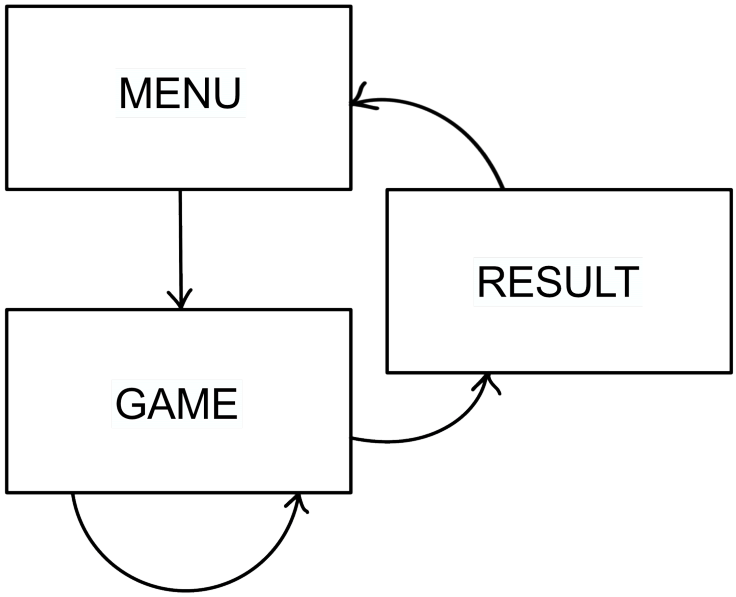
Questions

INTEGER	id	PK NOT NULL
TEXT	type	NOT NULL
TEXT	image	
TEXT	question	NOT NULL
TEXT	answers	NOT NULL
TEXT	correct	NOT NULL

Game

INTEGER	turn	PK NOT NULL
TEXT	board	NOT NULL

Site Map



Task Breakdown

TASKS	DEVOS	EXPECTED COMPLETION DATE
FEF	Everyone (Mainly Joyce)	12/22/25
Animations	Joyce Lin	12/11/2025
Flask	Evan Khosh	12/15/25
Database	Evan Khosh	12/15/22
Questions (interact with APIs)	Sean Zheng	12/18/25
Chess logic	Zixi Qiao	12/17/25

Chosen FEF: Bootstrap

- We plan on using Bootstrap for the majority of our FEF. Specific features we'd like to make use of are alerts, modals and flex grid (subject to change). Modals would be the most significant inclusion as it allows us to implement our trivia feature without having to make a completely new page.
- The reason why we are choosing *not* to use Foundation is because it lacks many unique features that make it attractive that Bootstrap does not offer (though its documentation

was the easiest to go through). Regarding Tailwind, despite it being interesting, most of its features are locked behind a paywall, thus making it generally an annoyance to work with (opening things and hoping that it's available for free).

Extra components

- Animated pieces
- Music
- Pieces fly away when spot taken (spinnn)
- Menu that allows you to choose what categories that you want to include
    - Select difficulty
        - Differing timer times
        - SINGLEPLAYER: how smart the bot it is
    - Select singleplayer vs multiplayer
        - Play against a friend locally or play against a Chess API
- Utilizing the date/time library, make the menu background change depending on whether it is night or morning (day —> black, night —> white)
    - Use existing walking animations and have them walk across the screen one by one
    - Have a toggle that allows you to reverse the time reqs —> night become day and day become night
        - Allows for testing at different times of day

Piece Number Identifiers:

- White - Positive
- Black - Negative
- Empty - 0
- Pawn - 1
- Knight - 2
- Bishop - 3
- Rook - 4
- Queen - 5
- King - 6