Playtesting Criteria:

- Level Failure
    - Resets score
    - Resets timer
    - Resets obstacle
- Powerups
    - Do as they say
    - Expire at the correct time
    - Varieties:
        - SpeedUp
        - JumpBoost
        - Clock
        - Shield
- Level Success
    - Saves level progress to correct user
    - Allows access to next level
    - Score Is recorded to leaderboard
- General
    - Obstacles respawn when they go off screen
    - Levels are completable
- Menus
    - Every button does what it says it does
    - Varieties:
        - Main Menu
        - Pause Menu
        - Level Failed Menu
        - Level Complete Menu

Playtesting results:

- Level Failure:
    - Level failure does reset all necessary items to zero and reset the positions of all obstacles
- Powerups:
    - Speedup did not actually speed up the level on first playthrough, just the level backgrounds. This was patched immediately.
    - JumpBoost, clock, and shield worked as intended.
- Level Success
    - Level progress is saved
    - Scores are recorded in leaderboard

        - Next level is accessible
    - General
        - Obstacles respawned in most cases, except for rocks in level 1 under certain conditions.
            - Rocks in level 1 would fall off the edge of the map if the page was too zoomed in. This was patched by removing gravity from the rocks
        - Level 1 was completable in the present state
        - Level 2 was incredibly difficult to complete without relying on the luck of the powerups. To remedy this, the speed and size of the boulder was reduced to give the player more time to react.
        - Level 3 was basically impossible to complete without relying on the shield power up. To remedy this, the hawks were unfortunately removed, as there didn't seem to be a way to incorporate them while keeping the level completable.
    - Menus
        - Each level button in ChooseLevel works correctly
        - The go back button had an issue relating to its nature of returning to the previous page. If the user were to "Quit Level" and then click "Go Back" on the subsequent page, they would be returned to the level they just left. This was patched immediately.
        - The Main Menu buttons all work as intended.
        - The Pause Menu buttons all work as intended.
        - The Level Failed Menu buttons all work as intended.
        - The Leve Complete buttons all work as intended.

Backend/API Criteria + Results

Criteria	Results
Flask App setup.	
1. Virtual environment setup loads all required packages from "requirements.txt" - Verify **visual** because we don't get any ImportErrors	Pass (no pytest required)
2. `flask run` starts up with the expected message "Running on localhost:5000" message and no errors.	Pass (no pytest required)
3. `flask init-db` creates the `users` and `leaderboard` table as specified in "schema.sql" without errors. - Verify in sqlite3 display i.e. vscode extension	Pass (no pytest required)

Endpoints	
“GET /load_users” returns a json with a `users` key that has a dictionary value. - Case No Users: If the dictionary is not empty - Case (Otherwise) - it has the inner keys as the columns in the `schema.sql` corresponding to the username and levelReached. -	Pytest file: /test_load.py Case No Users: Pass Else: Pass
“POST /add_user?username=[username]”, Adds a user to the database and returns json with successful message - Case valid username: if username is between 3 and 20 characters. - Case invalid username: if username is less than 3 characters or more than 20 characters.	Pytest file: /test_add.py Pass
“POST /increment_user_level?username=[username]”, When user completes a level, levelReached gets incremented. - Verify by creating a new user, incrementing their level twice to see it changes from 1 -> 2 and 2 -> 3 - Assumes username is valid, but also designed to return status 200 if the username is invalid.	Pytest file: /test_increment_level.py Pass
“POST increment_score? **username=[username]** **&score=[score]** **&levelScore=[lvlScore]”** Increments users score if their new score is higher than their previous one	Pytest file: /test_increment_score.py Pass
“GET /load_leaderboard” returns a json with all users and their total scores - Test: visual (LoadLeaderboard page)	Pass (no pytest required)
“GET /read_user_level” returns a json containing the level number that the user has reached.	Pytest file: /test_read_level.py Pass (pytest)