Hogwarts Hunt Quest

# Get started with Hogwarts Hunt Quest

# What you can do with Hogwarts Hunt Quest

* 1. Hogwarts Hunt Quest is a web application which features an interactive treasure hunt game. This game is designed to test player’s certain soft skills such as:
* Decision making
* Time management
* Critical thinking
* Creativity
* Persistence and perseverance

# Installation instructions

* 1. Minimum requirements
     1. **The latest version of your internet browser**. We recommend the following browsers:
        1. [**Google Chrome**](https://www.google.com/chrome/) (PC/Mac)
        2. [**Mozilla Firefox**](http://www.mozilla.org/en-US/firefox/new/) (PC/Mac)
        3. [**Safari**](http://www.apple.com/safari/) (Mac)

To check your browser version and upgrade:

* + - * Open your browser
      * Go to the main menu of your browser
      * Look for **Help**and/or **About** *<Browser Name>*
      * Click [**here**](https://zyngasupport.helpshift.com/a/zyngagames-com/?s=technical-help&f=how-do-i-find-my-browser-version) for more details.
    1. **A reliable internet connection**, as follows or better:
       1. DSL (speeds ranging from 128 Kbps to 8 Mbps)
       2. Cable (speeds ranging from 512 Kbps to 20 Mbps)
       3. Wireless Internet (speeds vary depending on the network)
       4. T-1 Lines (speeds ranging up to 1.544 Mbps)
    2. **Recommended Computer or Hardware Specs**
       1. CPU - Computer processor should be at least 4-Core.
       2. Graphics Card - At least 512MB Video Card.
       3. Memory (RAM) - At least 2GB of RAM.

# How to play

Click on Sign up and create your account using email Id.

Login using username and password provided at the time of signup.

Once the players have logged in successfully click on Start to start the game.

The players will be given a clue, upon which they must decide which place they want to visit on the map.

Players can click on locations to visit the places.

In each place, the players will have to answer a question in order to generate the next clue.

One can undo any answer and go to previous level.

Once the players have answered all the questions their total time taken will be noted.

Players can save and quit game and resume from same level and same time.

# Code Documentation

## Technologies used:

Main technology that was used to build this web application was Python where we used the Flask Framework and Javascript. HTML and CSS were also used to create templates and styles.

Requirements are as follows:

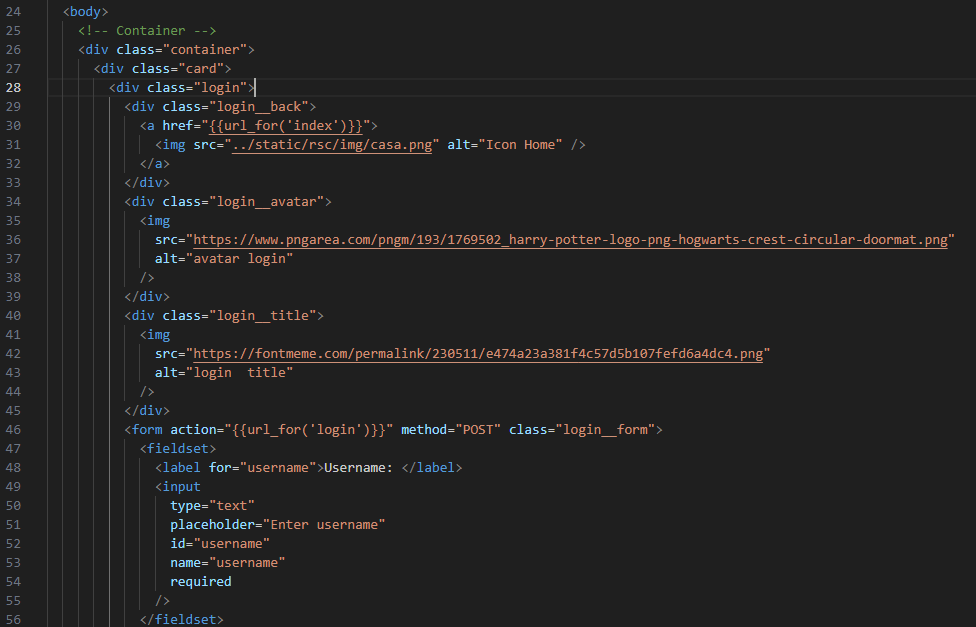
* blinker==1.6.2
* click==8.1.3
* colorama==0.4.6
* Flask==2.3.2
* Flask-MySQLdb==1.0.1
* importlib-metadata==6.6.0
* itsdangerous==2.1.2
* Jinja2==3.1.2
* MarkupSafe==2.1.2
* mysql==0.0.3
* mysql-connector==2.2.9
* mysql-connector-python==8.0.5
* mysqlclient==2.1.1
* Werkzeug==2.3.4
* zipp==3.15.0
  1. HTML
     1. Index.html

A screen shot of a computer program

Description automatically generated with low confidence

Description:

* This is the index.html file. This is the first page of the website. Linked to global-styles.css and home.css for styling the elements.
* The first line is a comment, indicating that the following code defines the header section of the website. The **header** element has a class of **header**.
* There is a the navigation menu section of the header, which is contained within the **header\_\_row1** division. It has a class of **nav2** and contains an unordered list with three list items, each with a link to a different page on the website. The links use Flask's **url\_for** function to generate the URLs based on the function names specified in the Flask application.
* Closing tags close the **header\_\_row1** and **header\_\_wrapper** divisions, as well as the **header** element.
* The second nested division in the header section, with a class of **header\_\_row2**. It contains a division with a class of **header\_\_img-title**, which has a heading element with a class of **main\_landing** and the text "Hogwarts Quest Hunt". Below the title is a button with a link to the login page, contained within a form element that uses the GET method to submit the form data.
  + 1. Login.html

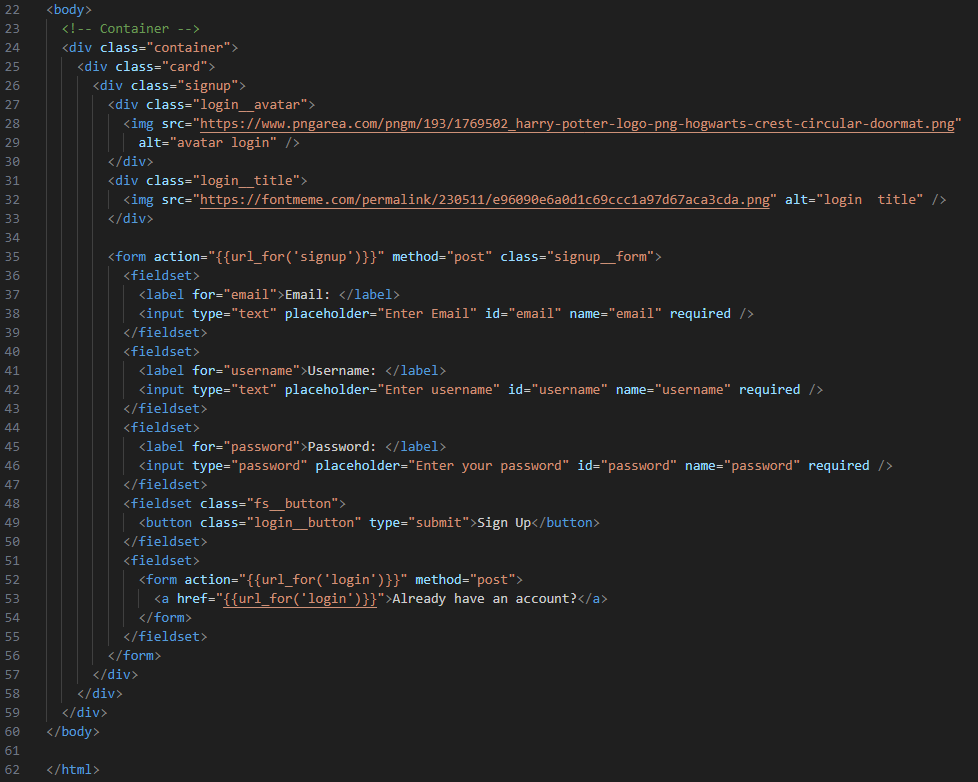


A screen shot of a computer screen

Description automatically generated with low confidence

Description:

* This code represents the login page of the website and is contained within a container element. The login page is designed to allow registered users to log in to the website with their credentials.
* The page starts with a header that contains a back button that links to the home page, and a title with an image of the Hogwarts crest. The header provides users with a way to navigate back to the home page if they need to.
* Below the header is the login form. The form includes two input fields for the user's username and password. Both of these fields are required, meaning the user must enter a value before submitting the form. Once the user has entered their login credentials, they can click on the login button to submit the form.
* If the user doesn't have an account, there is a link to the signup page. The link is located at the bottom of the login form and provides an easy way for users to navigate to the signup page if they need to create an account.
* Overall, the login page is designed to be simple and straightforward, with clear instructions and links to help users navigate the site.
  1. Signup.html



Description:

* The HTML code starts with the opening **<body>** tag.
* Inside the body, there is a **<div>** element with a class of "container", which serves as the main container for the page.
* Within the container, there is a **<div>** element with a class of "card", which contains the signup form.
* The signup form has a class of "signup\_\_form" and an action attribute that points to the server endpoint for handling signup requests.
* The signup form also contains three **<fieldset>** elements that group related form elements together.
* The first fieldset contains a label for the email input field and the input field itself, which has an ID of "email", a name of "email", and the "required" attribute set.
* The second fieldset contains a label for the username input field and the input field itself, which has an ID of "username", a name of "username", and the "required" attribute set.
* The third fieldset contains a label for the password input field and the input field itself, which has an ID of "password", a name of "password", and the "required" attribute set.
* The form also has a "Sign Up" button, which is a **<button>** element with a class of "login\_\_button" and a type of "submit".
* Below the signup form, there is another fieldset that contains a link to the login page for users who already have an account.
* The link is a **<a>** element with an href attribute that points to the server endpoint for the login page.
* The HTML code ends with the closing **</body>** and **</html>** tags.
  1. Startpage.html

A picture containing text, screenshot

Description automatically generated A computer code on a black background

Description automatically generated with low confidence

Description:

* The page includes a header section with a navigation menu and a logo.
* The navigation menu consists of links to the Home page, User Details, Rules, Leaderboard, and Logout.
* The page has a main title "Hogwarts Quest Hunt" and a subtitle that greets the user by their username.
* There is a "Start" button that takes the user to the game page when clicked.
* The page is designed with responsive layout and can be viewed on different devices.
* The code defines a JavaScript function named setCookie().
* The function takes three parameters: name, value, and days.
* The purpose of the function is to set a cookie in the user's browser.
* The value of the name parameter is used as the name of the cookie.
* The value of the value parameter is used as the value of the cookie.
* The value of the days parameter is used to set the expiration time of the cookie.
* If the days parameter is provided and is greater than 0, the cookie will expire after the specified number of days.
* If the days parameter is not provided or is less than or equal to 0, the cookie will expire at the end of the current session.
* The current\_time cookie is being set to the value of time template variable, which is expected to be a string representing the current time in a specific format.
* The -1 value for the days parameter is used to delete the cookie if it already exists.
  1. game.htmlA picture containing text, screenshot, software

     Description automatically generated A screen shot of a computer program

     Description automatically generated with medium confidence A screen shot of a computer program

     Description automatically generated with low confidence A screen shot of a computer program

     Description automatically generated with medium confidence

Description:

* The HTML code defines a page layout for a game with buttons, a timer, and score box.
* The code is wrapped in a <body> tag and contains a <main> tag as the main container.
* There is a <div> container with a class of "container" that contains all the game elements.
* Inside the container, there is another <div> container with a class of "buttons" that contains buttons for different locations.
* A <form> tag with an action of "cards" and method of "post" is defined that wraps the buttons for location selection.
* Each location button is an <input> tag with a type of "submit" and a value that represents the location name.
* The "Undo Last Answer" button is an <input> tag with a type of "submit", name of "prev-level", and value of "Undo Last Answer".
* The "Save & Quit" button is an <input> tag with a type of "button", an id of "quitButton", and a value of "Save & Quit".
* There is a <div> container with a class of "clue-box" that displays a clue for the current level.
* There is a <div> container with a class of "score-box" that displays the level and timer.
* The timer is defined in a <div> tag with an id of "timer" and an initial value of "00:00:00".
* The level is displayed in a <div> tag with a class of "level" and an initial value of "{{level}}".
* An <img> tag with a source of "../static/rsc/img/map.jpg" is included that displays an image of a map.
* This is a JavaScript script that performs several functions related to a timer. It begins by checking if there is an alert message to display, then it retrieves the previous timer value from a cookie. If the timer value exists, it starts the stopwatch with the previous value. Otherwise, it starts the stopwatch at 0.
* The script defines a startStopwatch() function that starts a setInterval timer, updating the time every 10 milliseconds. It formats the time into hours, minutes, and seconds, and sets the text content of an HTML element with the timer display. It then converts the time to total seconds and sets a cookie with the current time value and interval ID
* The script also defines a setCookie() function to set a cookie, and a getCookie() function to retrieve a cookie.
* Finally, the script defines a saveTimerValueAndQuit() function that saves the timer value in the database and navigates to the start page. It retrieves the timer value from the HTML element, makes an AJAX request to the server to save the timer value, and then navigates to the start page. The Quit button is given an event listener that calls saveTimerValueAndQuit() when clicked.
  1. Card.html

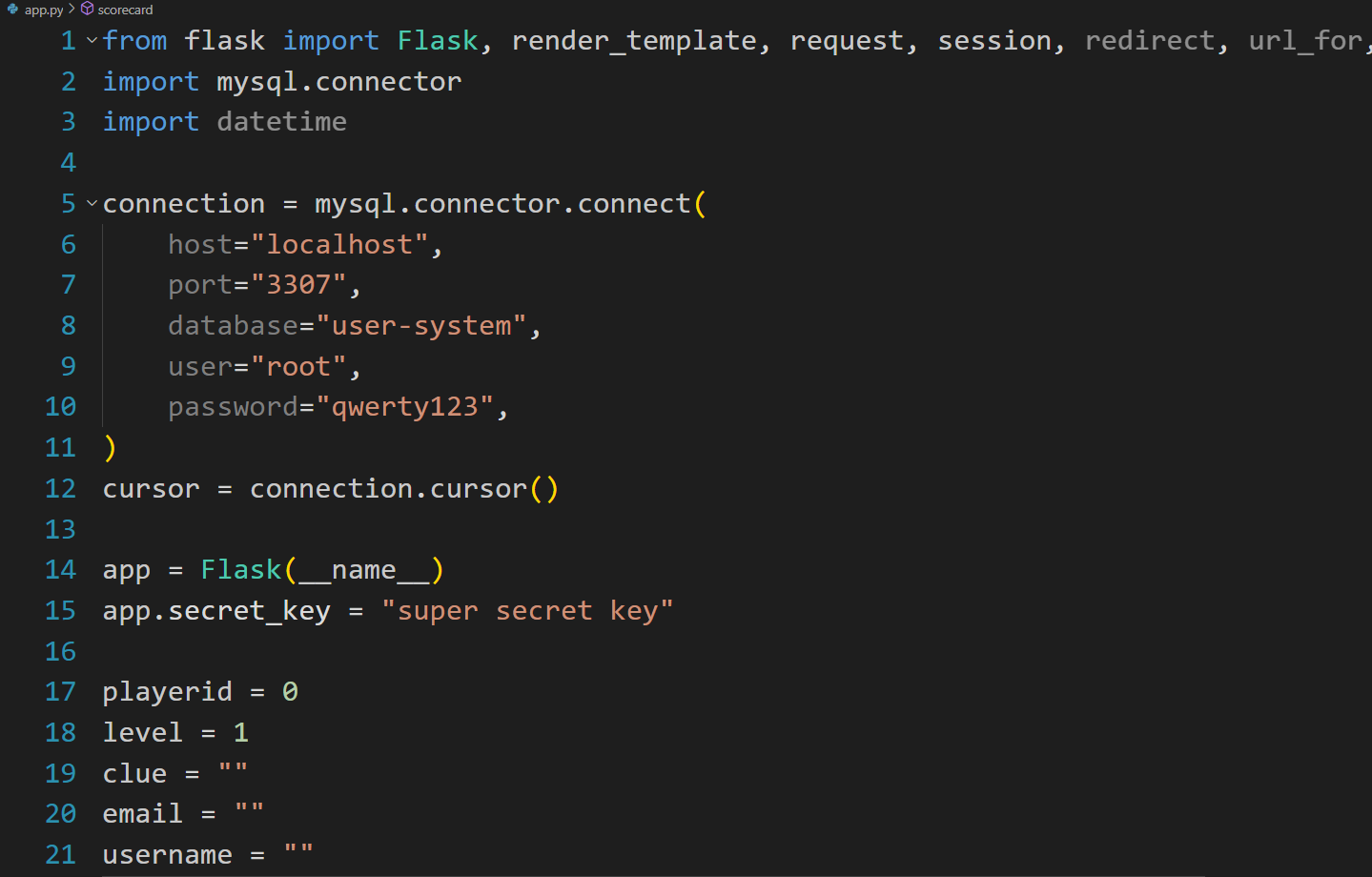
A picture containing text, screenshot

Description automatically generated

Description:

* This is the HTML code for a page that displays a timer, a question, an image, and a form to submit an answer.
* The timer is displayed in a div element with id="timer".
* The question is displayed in a div element with class="question".
* The image is displayed using an img element with src="{{ name }}".
* The form to submit an answer is using the POST method and action="{{url\_for('cards')}}".
* The form contains an input field with type="text" and id="answer", and a submit button with class="submit\_button" and type="submit".
* There is also a button to go back to the previous page, using a form with action="game" and a back button with class="back-button" and type="submit".
  1. App.py

Description



* The code is a Flask application that implements a user system and a game based on the Harry Potter theme.
* It establishes a connection with a MySQL database using the mysql.connector library.
* The application uses session-based authentication to manage user login and access control.
* It provides routes for different pages such as the index, login, about, start page, scorecard, user detail, signup, game, cards, save timer, and rules.
* The login route handles the user authentication by checking the provided username and password against the database records.
* The scorecard route retrieves user records from the database and calculates the time in a formatted manner to be displayed in the scorecard template.
* The game route retrieves a question from the database based on the current game level and renders the game template with the question as a clue.
* The cards route handles the game logic, including checking the user's answer, updating the game level, and rendering the appropriate template based on the chosen button or action.
* The save\_timer route receives the timer value from the frontend and updates it in the user's record in the database.
* The code includes several utility functions for converting timedelta to seconds and formatting time, as well as global variables to track the player's progress and current level.

**Conclusion:**

In conclusion, the Harry Potter treasure hunt game provides an engaging and interactive platform to test and develop various soft skills. Through the challenges, riddles, and clues based on the Harry Potter universe, players have the opportunity to exercise their problem-solving abilities, encouraging critical thinking and creativity. The game prompts players to pay close attention to details, fostering their observation skills and enhancing their ability to connect information. Persistence and perseverance are vital as players encounter obstacles and must persistently seek solutions. Additionally, collaboration and teamwork may be required, promoting effective communication and the ability to work together towards a common goal. Time management skills are put to the test, as players navigate through tasks within specific timeframes. Moreover, the game serves as a knowledge assessment, evaluating players' understanding of the Harry Potter series. Finally, the treasure hunt game demands attention, focus, and adaptability as players adapt their strategies to overcome unexpected twists. Overall, the Harry Potter treasure hunt game serves as an enjoyable and immersive platform that not only entertains but also effectively tests and enhances a user's soft skills.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |