BABEȘ-BOLYAI UNIVERSITY CLUJ-NAPOCA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

SPECIALIZATION COMPUTER SCIENCE

DIPLOMA THESIS

Exploring the world and interactive geographic learning through Google Street View locations based web application

Supervisor

Assoc. Prof. Molnar Arthur-Jozsef

Author

Rus Ilie Daniel

2024

Abstract

In order to promote geographic learning through the study of Google Street View sites, this bachelor thesis explores the conception, creation, and development steps of an interactive web application. Inspired by the popularity of an already-existing platform like GeoGuessr, the project`s goal is to give users an engaging and instructive experience. Driven by a personal interest in geography and a desire to use technology to promote learning, the project tries to offer an approachable and free platform for exploration, that has public source code.

In the introduction, we will discuss my motivation for choosing this thesis topic. Chapter two will cover web mapping services like Google Maps and Google Street View, which relate to my application. Chapter three will examine the state of the art in this niche, focusing on Geoguessr and its community, the benefits of similar games, and potential app optimizations. Chapter four will explore the app's potential and optimization strategies for the target problem. In chapter five, we will discuss the technologies used to build the application. Chapter six will detail the implementation of the backend and frontend. Chapter seven will present our conclusions, and chapter eight will outline potential future work. Finally, chapter nine will list all references used in the thesis. The conclusion reflects on the objectives and outcomes of the project, highlighting the importance of optimization, user engagement, and creating valuable educational resources for learners and enthusiats in the domain of geography.

In summary, this thesis represents a comprehensive exploration of the intersection between technology, education, and geography, with the ultimate goal of creating a valuable resource for learners and enthusiasts alike in the domain of geography.

Table of contents

1. Introduction

1.1. Motivation

One of the main interests that I have had during my lifetime has been geography. During the pandemic I happened to stumble upon a game called GeoGuessr[1], whose concept sparked my curiosity, at a peak time, in terms of representing what its community wanted from such a game. In it`s essence, the game challenges your ability of finding a random location on Google Maps, after you are dropped and allowed to check your surroundings and search for any type of clues in the Google Street View location you are on. After you save your guess, you will receive up to 5000 points[2], based on how close you were to the actual location. A lot of game modes have been developed up until today, including multiplayer ones, where you are allowed to compete against other people. During the last 4 years, I have been a regular user and enjoyer, may I add, of this web application, developing my skills of identifying certain unique characteristics and patterns for certain countries and regions, such as license plates, pole tops, bollards, writing styles etc[3]. In the meantime, Geoguessr has gradually transformed into a regular game, with a UI resembling of a console game and features such as avatars, growing concerns within the community as to whether the application is actually trying to address real issues like cheating in multiplayer game modes and bringing a simple game on the market in which the users are challenged to guess where on the map they are, instead of making a very complicated graphical interface.

1.2. Target problem

Therefore, my aim is to build a web application from scratch using PHP, JavaScript, HTML and CSS, that allows the user the possibility to try to find the location on the map where he is dropped randomly, until he decides he wants to return to the home page where he can see the menu. The user can choose to see the statistics of every user in the database and to see his own statistics, individually. The statistics a user will have access to are the total number of rounds played, the total score, representing the sum of all the scores a user receives for all of his guesses, the average score per round and the best score awarded for a singular round of guessing. The admins can also delete a user or promote another user to the admin role from a menu only available to the admins. The difference that my app will bring, most importantly is public source code, the game will be free, especially due to its lack of popularity which will not catalyse a rise the cost of the Google Maps API, that we will be using for the application. Also, the game will have a simple and easy interface to interact with, the sole purpose of the game being the challenge of trying to recognize the location

you are randomly dropped in and memorizing certain patterns which will help you in later rounds. The backend will be made in PHP, it is a simple, yet effective server scripting language that I enjoyed using during the course of Web Programming in my second year at university. For the frontend, I will use CSS, to put it very simply.

In the second chapter of the thesis, we will talk about the web mapping services, Google Maps[4] and Google Street View[5], as they are related to the implementation of my application. In the third chapter we will approach in a more nuance manner the state of the art in this niche area, which concisely leads us to the same subject of discussion as we did in the introduction: Geoguessr and its community and after that we will discuss data regarding the benefits of engaging in a game with a similar concept, and whether we can optimize the app or add any features to further enhance certain skills of our users. In the fourth chapter, we will discuss about the potential of the app and about the optimization and the proceedings in regards to the target problem. The fifth chapter of the thesis will represent the part in which we will talk about the technologies that are used for the purpose of building the application. In the sixth chapter we will delve into implementation of the backend and frontend of the application. We will draw the conclusions in the seventh chapter. Lastly, we will talk about the potential future work for this web application in chapter eight. The references of the articles, websites and every source used in the process of creating the app and the written thesis will be found in the ninth chapter.

1.3. Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used ChatGPT in order to improve the quality of the writing and provide several initial ideas that were developed by me during the creation of this thesis. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the thesis.

2. Related work – web mapping services

In this chapter, we will go through two of the web mapping services, which revolutionized the way we interact with geographical data and the access at navigating the world from the comfort of our gadgets.

2.1. Google Maps

Google Maps represents of the most important innovations in the history of technology. It is a web mapping service developed by Google that offers users maps, terrain and satellite imagery, traffic and public transport information and route planning tools. Google Maps facilitates the everyday navigation in every city and every country in the world in which it is available, it can provide you the most appropriate route from one point of the city to another. Additionally, Google Maps offers information about the public transport and the traffic, and very accurate navigation for both driving and walking routes. The algorithms, techniques and technology used by Google Maps are highly advanced, thus representing an important part of our lives[6]. There is documentation for the code used on Google Maps website, which helped during the development of the application[7].

2.2. Google Street View

Google Street View, a component of Google Maps, offers users street-level imagery of its locations worldwide. Developed by Google, this feature enables users to virtually explore streets, neighbourhoods, and landmarks with panoramic photographs captured from specially equipped cars. Beyond streets, Street View extends its reach to include indoor spaces such as museums, businesses, and cultural sites, allowing users to experience these environments in a virtual reality.

3. State of the art in virtual geographic discovery games

As previously mentioned in the introduction, in this field Geoguessr is definitely the main virtual geographic discovery game on the market, having by far the best and most well known product in the industry. Therefore, when discussing the state of the art in our area, we will cover Geoguessr to get a better understanding of what it means, the potential benefits of playing such a game[8], its community and the current situation of the game.

3.1. Geoguessr

GeoGuessr is a web application which represents a geography game in which players are tasked to guess a random location from Google Street View imagery. Despite its simple premise, GeoGuessr serves as both an entertaining and learning tool for geography enthusiasts. Players can learn and identify patterns in characteristics of a region such as writing systems, architecture, driving practices, flags, vehicle license plates, poles, pole tops, bollards and many others.

Gameplay primarily revolves around the normal mode, where players guess locations displayed in five rounds of a random street view location. There are other game modes like Battle Royale, Duels, Streaks, Explorer Mode, Team Duel and, Maprunner.

The game's interface in the normal mode has Google Street View imagery, a map on which the user has to register a gues and a compass. Players can move around, pan or zoom onto the imagery, but they also have the option to disable one of the features for increased difficulty. An inset map allows players to place a pin to make their guess.

Reception of GeoGuessr has been positive, with the application quickly becoming viral after its release in 2013 and experienced renewed interest during the pandemic[9]. It has been appreciated for the educational and entertainment value. Notable content creators such as Rainbolt or Geowizard have contributed to its popularity through social media platforms like YouTube, TikTok, and Instagram.

To locate ourselves in the midst of the most discussed current subjects, artificial intelligence and its implication on our future, I will mention that in 2023, researchers at Stanford University developed a machine learning tool that was able to locate 40 percent of GeoGuessr locations with

an accuracy of 25 kilometres or better[10]. After three attempts, the model was able to beat Rainbolt, one of the most highly regarded players within the community.

In 2023, GeoGuessr hosted its first single-player in-person esports tournament, the GeoGuessr World Cup, featuring participants from around the world. The event showcased the game's competitive potential and further suggested that the relatable and polyvalent nature of Geoguessr could attract more users to it or a similar application.

3.2. Data regarding the benefits of engaging in virtual geographic discovery games

Introducing GeoGuessr into the General Physical Geography course at Mugla Sıtkı Kocman University in Turkey had a positive impact on student engagement and learning outcomes.[11] The study involved 103 out of 140 students who played the game, totalling 629 times. we will make a quick breakdown of the statistics that are available as a result of the study.

1. Participants:

- 140 students were asked to play games as an extracurricular activity for the geography course.

- 103 out of the 140 students participated in the study.

- These 103 students played the game a total of 629 times.

2. Game Participation:

- 74% of the participants had played the game.

- 16% had never played the game.

- 7% did not respond.

- 3% admitted to cheating in the game.

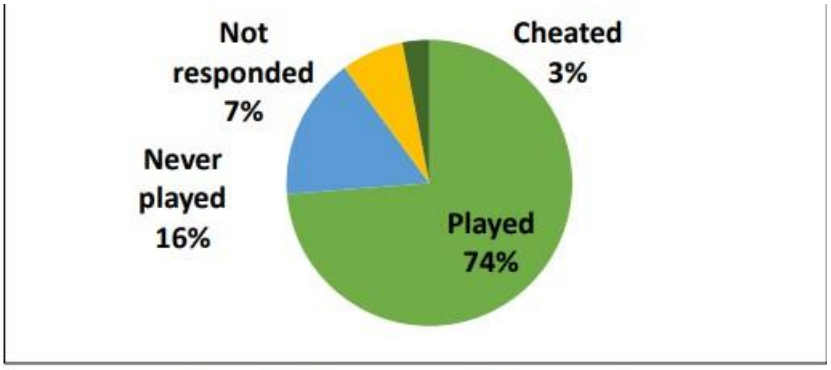


Image 2. Information regarding participants

3. Reasons for Non-Participation:

- 23 students had never played the game before.
- 10 students left questions unanswered or shifted topics.
- 4 students admitted to cheating to achieve perfect scores.

4. Indicators Used:

- Students used a total of 317 indicators across seven categories.

5. Indicators by Category:

- Chart/Plate: 30%
- Vehicle/Road/Traffic: 18%
- Flora: 17%
- Settlement/Architecture: 11%
- Terrain: 10%
- Climate: 9%
- People/Clothes: 5%

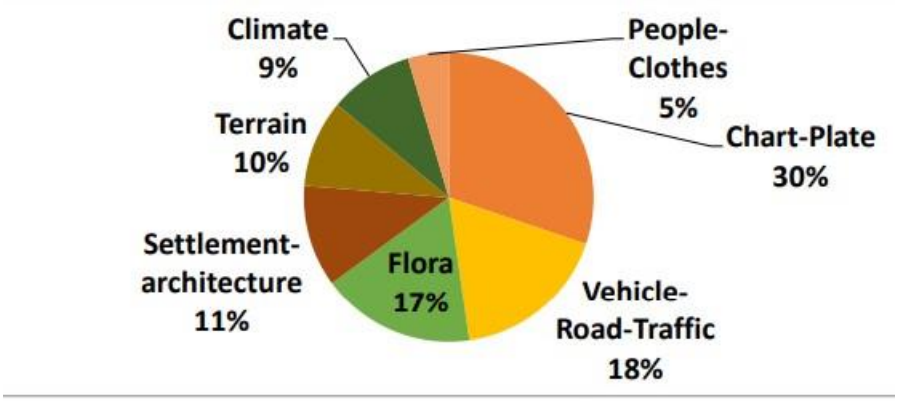


Image 3. GeoGuessr game features

6. Feedback from Participants:

- 40 participants believed that the games supported learning and teaching.

- 33 participants enjoyed their time playing the game.

- 18 participants said that they recognized the world and countries better.

- 17 participants reported improvements in their knowledge of location, place, direction, space, and maps.

- 5 participants mentioned improvements in their skills of association and interpretation.

7. Overall Impressions:

- 75% of participants gave positive impressions about the game.

- 25% showed less interest due to either technological restrictions or personal gaming habits.

8. Comparison with Previous Research:

- The findings align with previous research conducted in Bulgaria and the USA.

- Similar benefits were observed, including improvements in map reading skills, problem-solving abilities, and autonomous learning.

As a conclusion, integrating games like GeoGuessr into geography courses or your daily routine can improve learning outcomes, knowledge of locations and directions, understanding of maps and problem solving abilities.

4. Proposed approach

In this chapter, we outline the methodology and steps taken to develop "Locaisseur", our application. Our approach integrates various web technologies, including PHP, CSS, HTML, and JavaScript, with the data being stored in the MariaDB database managed via phpMyAdmin. Initially hosted on a local server using XAMPP, the application allows users to register, log in, and play an unlimited number of GeoGuessr rounds while tracking their statistics. Administrators have additional capabilities to manage users by promoting them to admin or deleting their accounts. This chapter details the potential features considered, the optimization strategies involved and the new score algorithm that the user receives after making his guess.

4.1. Potential in development

"Locaisseur" represents an innovative yet accessible and free approach to delivering the user a geographical location guessing web application, eliminating non-essential elements such as avatars. By employing widely recognized web technologies such as PHP, CSS, HTML, and JavaScript, the development process remains efficient, providing interactivity, functionality, structure and styling to web pages and applications. Utilizing MariaDB through phpMyAdmin ensures robust database management[12], allowing good handling of user data and game statistics.

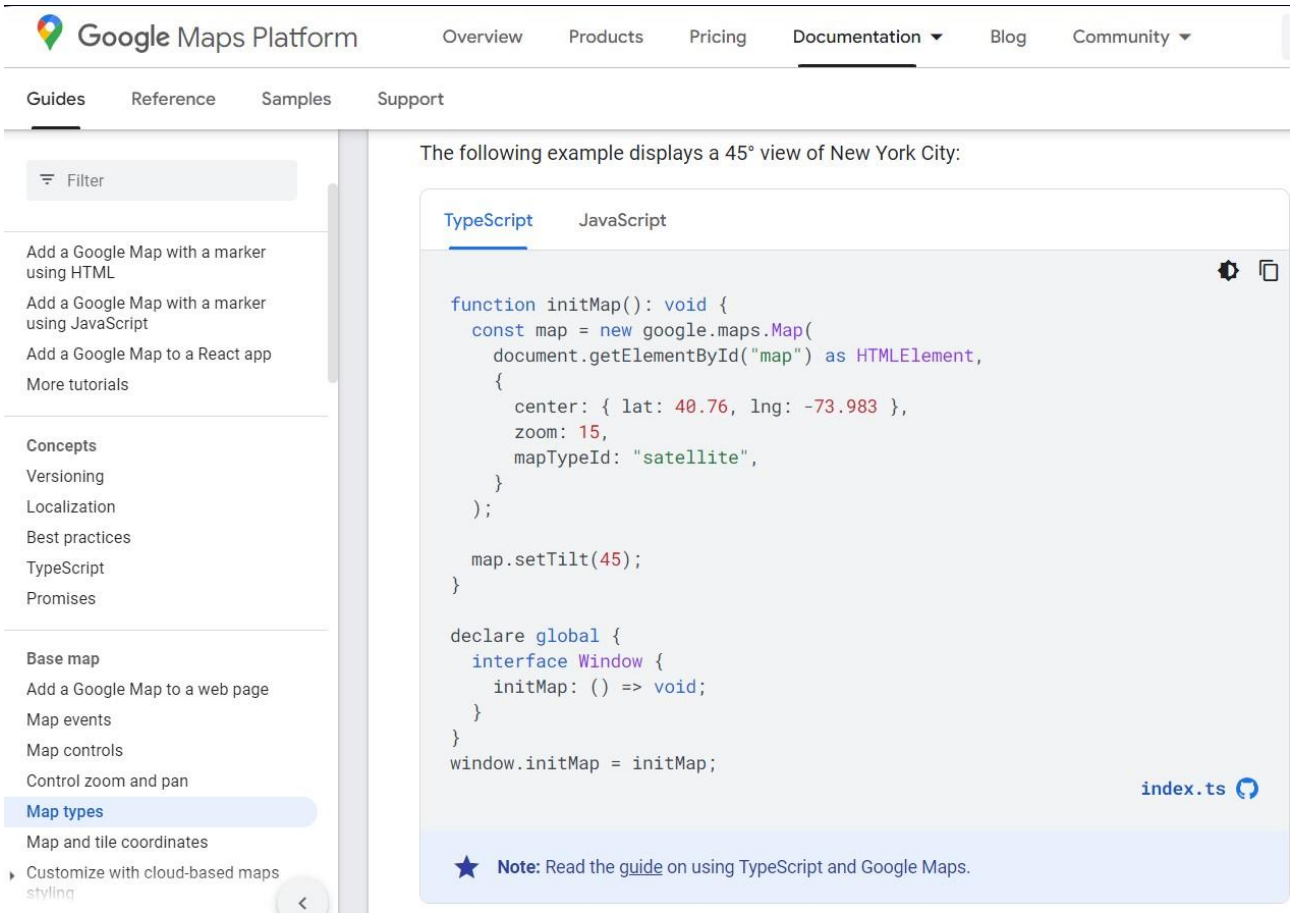
The simplicity of the interface and user experience ensures incoming interest from both casual players and geography enthusiasts, resulting in a broad audience. The structured, relational database management of MariaDB allows scalability and reliability as the user base grows.

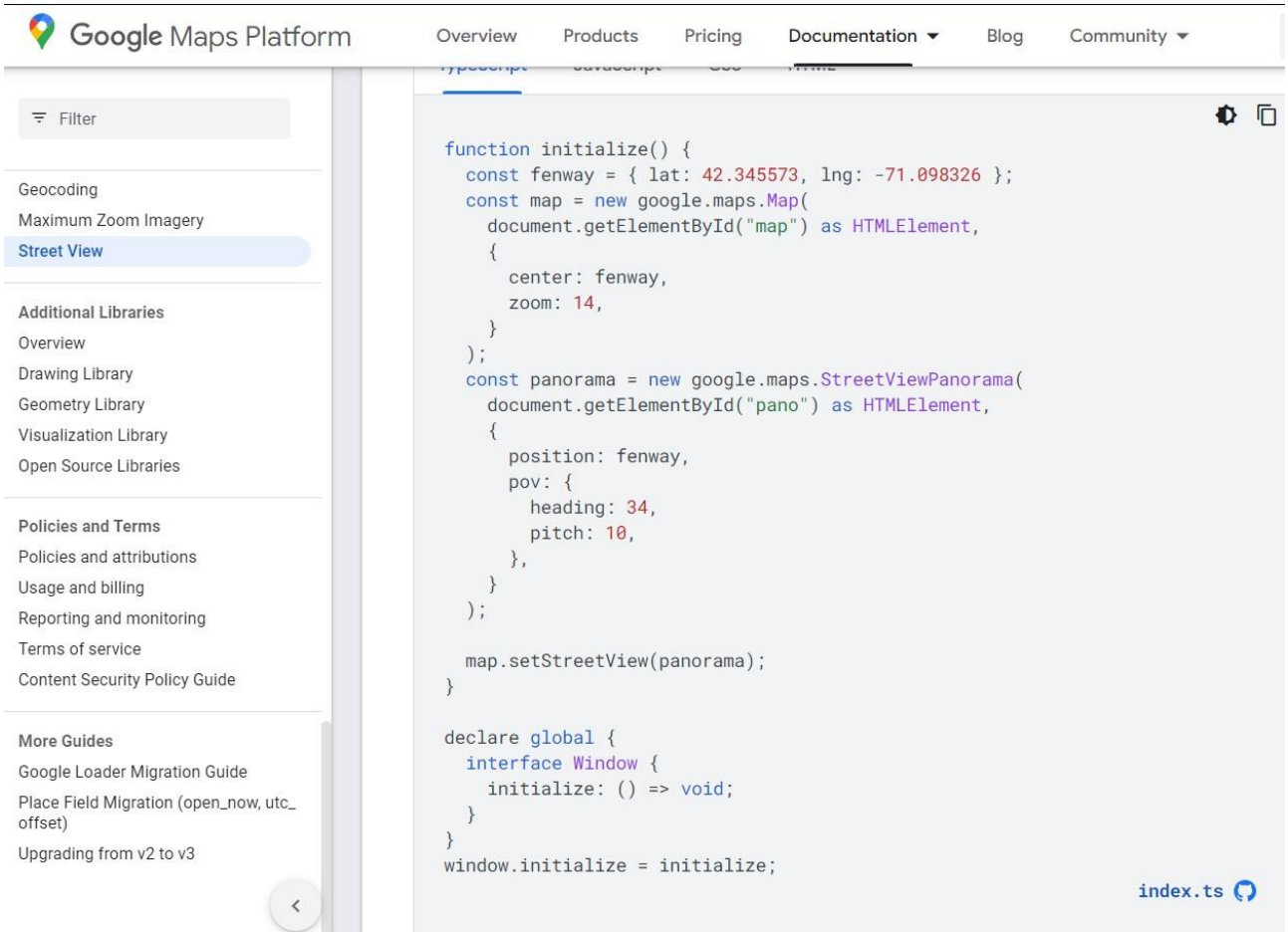
Designed with future deployment in mind, the application can transition from local hosting to a live server environment, expanding its reach and testing capabilities. Features such as overall statistics and user profiles will provoke community engagement.

4.2. Optimization and approaching the target problem

As we have already discussed several times already, the purpose of this application is to offer the user a straightforward and unpretentious option of exploring the world through Google Street View and testing their knowledge of the geographical locations. To achieve this, the first step was to acquire an API key in order to access the Google Street View imagery and Google Maps[13].

The next step was getting familiar with the documentation that Google is offering to manipulate the earth map and the street view imagery[14]. Here are two examples that were extremely helpful in displaying the map and the street view in my web application:





The most complex issue that needed to be tackled is the algorithm of calculation of the distance between the guess that the user inputs and the target he needs to hit in order to receive the maximum amount of points. The distance between the user's guess and the actual location is calculated using the Haversine formula[15]. This formula determines the shortest distance over the earth's surface, giving the precise distance between two points, ignoring any landform that would impact the distance travelled by a person during a trip or expedition. The implementation involves several key steps within the checkDistance function.

Firstly, the latitude and longitude of both the user's guess and the actual location are retrieved. These coordinates are represented by lat1, lng1 for the user's guess and lat2, lng2 for the actual location. The radius of the Earth, approximately 6371 kilometers, is stored in the variable R. Since latitude and longitude values are in degrees, they must be converted to radians. This conversion is done using the deg2rad function, which multiplies the degree value by Math.PI / 180. Next, the differences in latitude and longitude between the actual location and the guess are calculated and converted to radians, represented by dLat and dLng. The Haversine formula is then applied to these

differences. It involves calculating the value a using the sine and cosine of the latitude and longitude differences. Specifically, a is computed as the square of the sine of half the latitude difference plus the cosine of the latitudes multiplied by the square of the sine of half the longitude difference. The angular distance in radians, c, is calculated using the arctangent function, Math.atan2, of the square root of a and the square root of 1 - a.

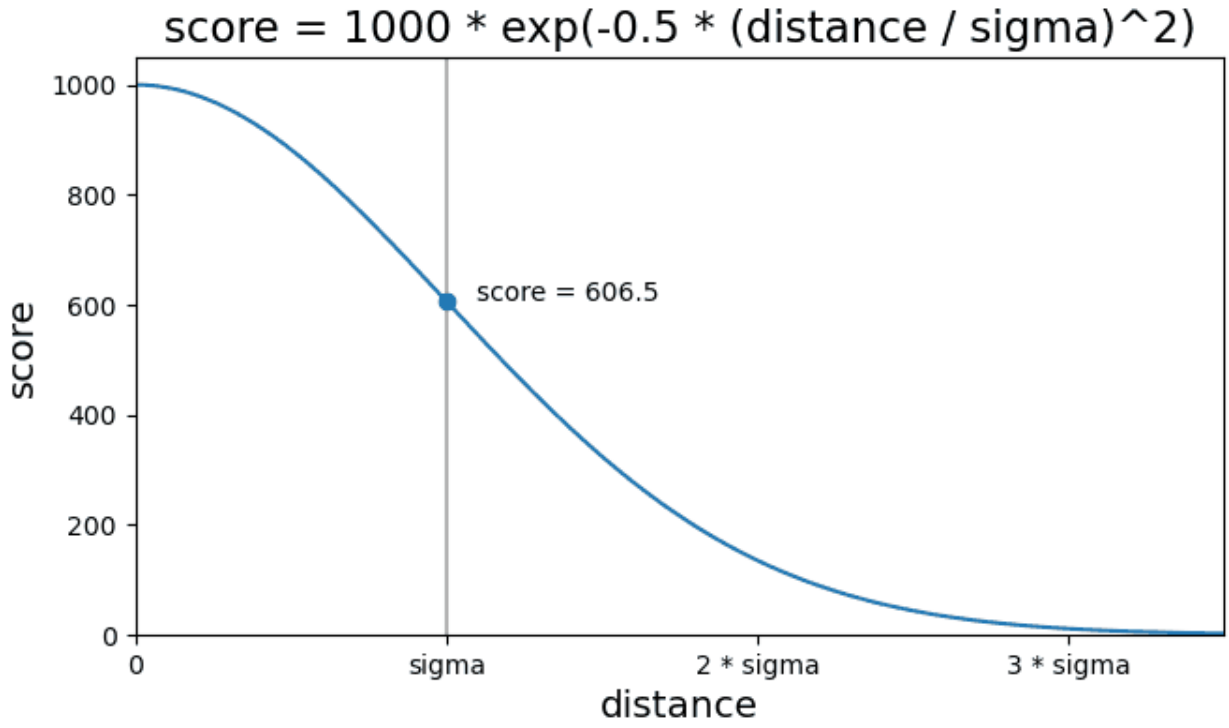
Finally, the distance is calculated by multiplying the Earth's radius R by c, yielding the distance in kilometers. This distance calculation is integral to the gameplay of "Locaisseur." After a user places their guess on the map, the checkDistance function is called to determine how close the guess is to the actual location. This distance is then used to score the user's guess.

Here is the code implementation:

```
function checkDistance() {
  var lat1 = guessMarker.getPosition().lat();
  var lng1 = guessMarker.getPosition().lng();
  var lat2 = currentCords.lat;
  var lng2 = currentCords.lng;

  var R = 6371;
  var dLat = deg2rad(lat2 - lat1);
  var dLng = deg2rad(lng2 - lng1);
  var a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
          Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2)) *
          Math.sin(dLng / 2) * Math.sin(dLng / 2);
  var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  var distance = R * c;
  return distance;
}

function deg2rad(deg) {
  return deg * (Math.PI / 180);
}
```

In "Locaisseur" the player's score is calculated based on the distance between their guess and the actual location. This scoring system awards more points for closer guesses. The process involves using the previously calculated distance and a mathematical formula to determine the score.

Firstly, the distance between the guess and the actual location is obtained by calling the `checkDistance` function. This function returns the distance in kilometers, which is then logged to the console for debugging purposes. The score is calculated using an exponential decay function defined by the following formula that will result in scores of the form below[16]:



In this formula, sigma is a constant that determines the rate of decay, set to 250 in our case. The exponential decay function ensures that the score decreases rapidly as the distance increases, reflecting the decreasing accuracy of the guess.

Here's how the scoring formula works in detail: the distance is divided by sigma, normalizing the distance relative to the decay rate. This normalized distance is squared, amplifying the effect of larger distances. The negative half of this squared value is taken, and the exponential function exp() is applied to it. The exponential function decreases rapidly for larger values of distance. Finally, the result is multiplied by 1000 to scale the score to a range between 0 and 1000, where a perfect guess (distance of 0) awards a score of 1000.

Here we have the code implementation:

```
  // Calculate distance
distance = checkDistance();
console.log('Distance calculated:', distance);

// Calculate score based on distance
const sigma=250;
score = 1000 * Math.exp(-0.5 * (distance / sigma) ** 2);
console.log('Score calculated:', score);
```

Subsequently the scores awarded for each round will be used to update the user`s statistics in the database. Once the score is calculated, the updateStatistics function is called with the score as its parameter. The updateStatistics function sends a POST request to the update_statistics.php script using the Fetch API. This request includes the score in a JSON format. The PHP script on the server side handles the update of the user's statistics in the database. Specifically, it updates the total rounds played, total score, average score, and best score for the user. The server responds with a JSON object indicating the success or failure of the update, which is logged to the console for debugging purposes. After updating the statistics, an informational window is displayed to the player, showing the target location, the distance from their guess to the actual location, and their score for the round. This is achieved using the Google Maps JavaScript API's InfoWindow class.

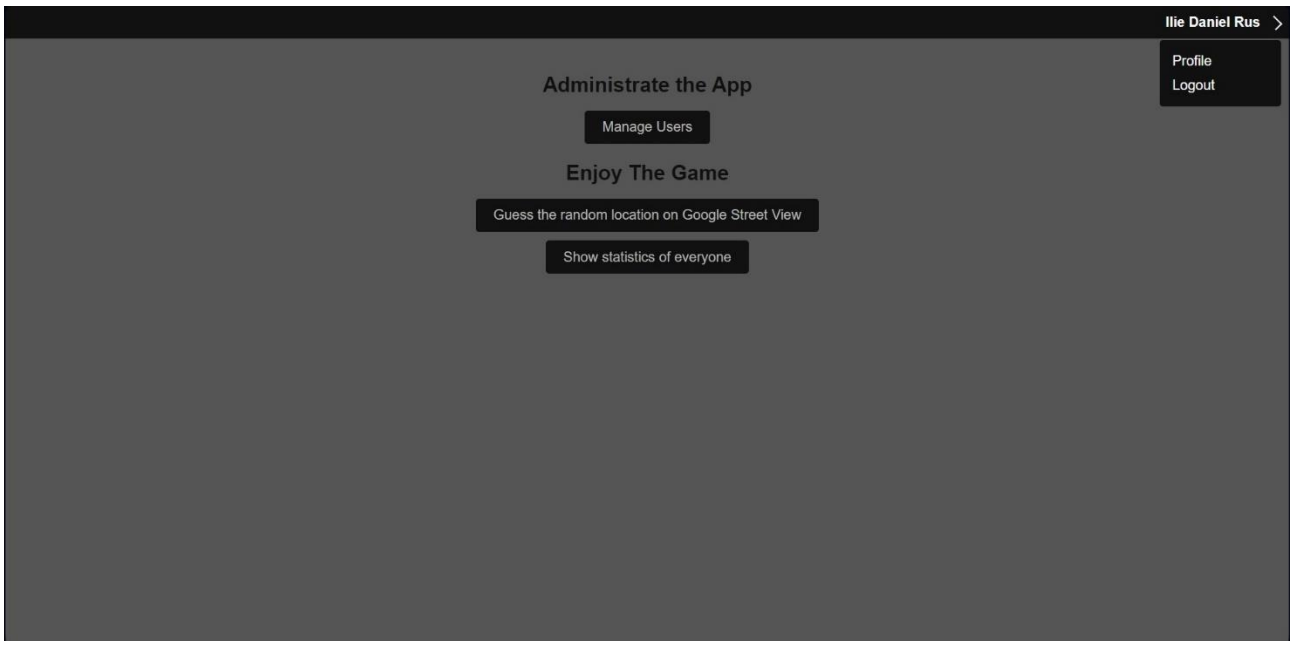
```
// Update user statistics in the database
updateStatistics(score);
console.log('User statistics updated');

// Display info window
const infoWindowContent = "Your target was " + currentCity + "<br>You were " + distance.toFixed(1) + " km away" + "<br>Your score for this round
const infowindow = new google.maps.InfoWindow({
  content: infoWindowContent,
});
console.log('Info window displayed');
```

```
function updateStatistics(score) {
  // Update roundsplayed, totalscore, avgscore, and bestscore in the database
  fetch('update_statistics.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      score: score,
    }),
  })
  .then(response => response.json())
  .then(data => {
    console.log('Success:', data);
  })
  .catch((error) => {
    console.error('Error:', error);
  });
}
```

The update_statistics.php script is responsible for updating the user's gameplay statistics in the database we are using. When the user's score for a round is received via a POST request, the script retrieves the user's username from the session. It then constructs an SQL query to increment the total number of rounds played, add the new score to the total score, recalculate the average score, and update the best score if the new score is higher. This query is executed against the database. If successful, a JSON response indicating success is returned; otherwise, an error message is returned, detailing any issues encountered during the update process. This ensures that the user's performance is accurately tracked and stored in the database after each round.

```php
// Get user's username from the session or request, assuming you have it available
$username = $_SESSION["username"];

// Update user statistics
$sql = "UPDATE user
        SET roundsplayed = roundsplayed + 1,
            totalscore = totalscore + '$score',
            avgscore = totalscore / roundsplayed,
            bestscore = CASE
                            WHEN '$score' > bestscore THEN '$score'
                            ELSE bestscore
                        END
        WHERE username = '$username'";

if ($conn->query($sql) === TRUE) {
    echo json_encode(array("message" => "User statistics updated successfully"));
} else {
    echo json_encode(array("error" => "Error updating user statistics: " . $conn->error));
}
```

As we have discussed earlier, this application has plenty of features to be improved upon or added, in order to generate the best possible experience for the user. Firstly, the interface can be refined by adding some high quality backgrounds. At this moment, we went for a simple dark mode background for the interface.



Secondly, we could generate the locations via an algorithm in the future instead of retrieving them from a hard coded list, it will provide a better experience as it will be more probable for the user to get a different location everytime. Two other points of discussion could be the deployment and improving the security of the web application. Deployment is necessary if we want to reach the general public, but also the lack of security could then become a problem. The solution to this will be the authentication with JWT tokens[17]. More on the subject of future work and optimization of the application will be discussed in chapter eight, named Future work.

5. Technologies used

This chapter will provide an insight onto the technologies that have been used to create the application “Locaisseur”, their role in creating the app and the advantages of using precisely them. PHP (Hypertext Preprocessor) serves as the server-side scripting language for "Locaisseur", facilitating the creation of adaptive content, manages database interactions, and handles session management. Here is an example of PHP allowing only users who are logged in to play the game, in order to be able to keep track of the users` statistics.

```php
<?php
$server = "localhost";
$username = "root";
$password = "";
$dbname = "licenta";

session_start();

if (!isset($_SESSION["username"])) {
    header("Location: index.php");
    exit();
}

// Create a connection
$conn = new mysqli($server, $username, $password, $dbname);

$username = $_SESSION["username"];
$isadmin = $_SESSION["isadmin"];
$realname = $_SESSION["realname"];
?>
```

HTML stands for Hypertext Markup Language, and CSS stands for Cascading Style Sheets. HTML establishes the structure of "Locaisseur," specifying the arrangement and hierarchy of material on web pages, while CSS improves presentation and styling to guarantee a coherent and aesthetically pleasing layout of the web pages, which will provide the users a good experience on the application[18].

Here is an example of displaying the menu on the main page, using HTML and CSS:

```php
<div class="menu">
    <?php if ($isadmin == 1): ?>
        <h2>Administrate the App</h2>
        <ul>
            <li><a href="manage_users.php">Manage Users</a></li>
        </ul>
    <?php endif; ?>
    <h2>Enjoy The Game</h2>
    <ul>
        <li><a href="classical_game.php">Guess the random location on Google Street View</a></li>
        <li><a href="show_stats.php">Show statistics of everyone</a></li>
    </ul>
</div>
```

```css
.menu {
    padding: 20px;
    max-width: 600px;
    margin: 0 auto;
    text-align: center;
}

.menu h2 {
    margin-bottom: 10px;
    color: #111;
}

.menu ul {
    list-style: none;
    padding: 0;
}

.menu ul li {
    margin-bottom: 10px;
}

.menu ul li a {
    display: inline-block;
    background-color: #111;
    color: #bbb;
    text-decoration: none;
    padding: 10px 20px;
    border-radius: 4px;
    transition: background-color 0.3s ease;
}

.menu ul li a:hover {
    background-color: #333;
}
```

JavaScript is the client-side scripting language used in "Locaisseur," allowing dynamic content updates, interactive features, and asynchronous server connectivity. Developers can create responsive web applications with improved user interaction due to its wide array of tools and frameworks and its versatility. JavaScript was the main pillar of the primary functionality of my app as it represents the vast majority of the code inside “classical_game.php”, which allows the user to see the Google Street View imagery, see the map on which they make the guess. All of user`s interactions are recorded and worked with through JavaScript.

MariaDB: The foundation of "Locaisseur's" data storage and retrieval system is MariaDB, an open-source relational database management system with significant power. It is the best option for efficiently and dependably managing user accounts, game data, and statistical information because of its strong performance, scalability, and support for SQL standards. Here is the table user that we work with in the database:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	**username**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
3	**password**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
4	**realname**	varchar(255)	utf8mb4_general_ci		Yes	USER			Change Drop More
5	**isadmin**	bit(1)			Yes	0			Change Drop More
6	**roundsplayed**	int(11)			Yes	0			Change Drop More
7	**bestscore**	int(11)			Yes	0			Change Drop More
8	**avgscore**	decimal(10,2)			Yes	0.00			Change Drop More
9	**totalscore**	int(11)			Yes	0			Change Drop More

The user has an auto incrementable id that he does not have to work with on the interface, a unique username, a password, a real name field that we display instead of the username, to improve the user experience, a field that determines whether the user is an admin or no, and four fields that represent this statistics, that we already discussed.

6. Application

In this chapter I will present the application “Locaisseur”, that we have discussed about up until now.

6.1. Specification

In this subchapter, we outline the methodology and steps taken to develop "Locaisseur," our application. Our approach integrates various web technologies, including PHP, CSS, HTML, and JavaScript, with the data being stored in the MariaDB database managed via phpMyAdmin. Initially it is hosted on a local server using XAMPP. The application allows users to register, log in, and play an unlimited number of rounds of guessing a random location on Google Street View while tracking their statistics, which they are able to see. Administrators have additional capabilities to manage users by promoting them to admin or deleting their accounts. This chapter details the implemented features considered, and the new score algorithm that the user receives after making their guess.

The intention is to offer a user friendly experience, and to keep the essence of the game, by not involving any non geography related matters on the application, while also keeping the code free source on Github.

6.2. Requirements

In the following subchapters we will outline the general criteria and conditions that needed to be met in order to consider the development of this application successful.

6.2.1. Functional requirements

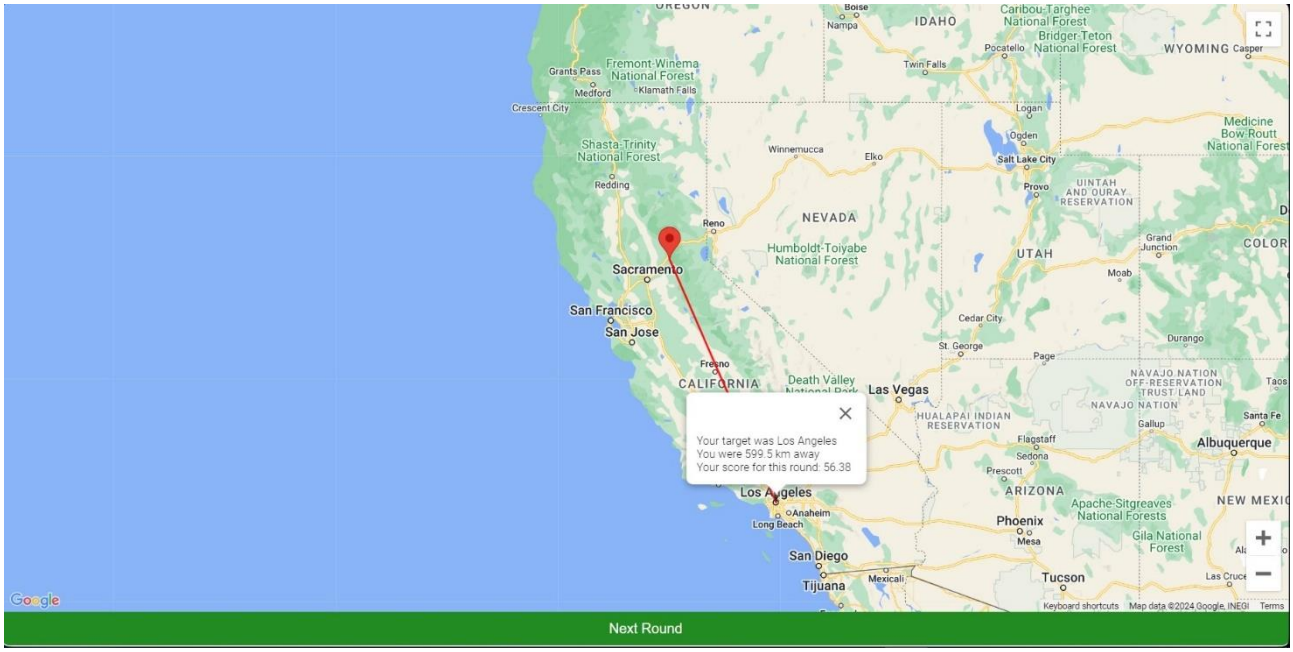
An integral part of the application is the possibility of registering a new user and implicitly the possibility to login, as well. If a user does not have an account, he will have to enter a username, his real name (or the name he wants displayed in the application) and a password. If the username chosen is unique the new user will be created and they will be logged in and sent to the main menu.

```php
// Process form submission
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Retrieve form data
    $username = $_POST['username'];
    $queryusernameunique="SELECT * FROM user WHERE username='$username'";
    $resultusernameunique=$conn->query($queryusernameunique);

    $password = $_POST['password'];
    $realname = $_POST['realname'];

    if($resultusernameunique->num_rows==0){
        $sql = "INSERT INTO user (username, password, realname) VALUES ('$username', '$password', '$realname')";
        if ($conn->query($sql) === true) {
            $_SESSION["username"] = $username;
            $_SESSION["realname"] = $realname;

            $querycurrentisadmin = "SELECT isadmin FROM user WHERE username = '$username'";
            $resultcurrentisadmin = $conn->query($querycurrentisadmin);
            $row = $resultcurrentisadmin->fetch_assoc();
            $current_isadmin = $row['isadmin'];

            $_SESSION["isadmin"] = $current_isadmin;
            header("Location: main_page.php");
            exit();
        } else {
            header("Location: register.php");
            exit();
        }
    }
    else{
        header("Location: register.php");
        exit();
    }
}
```

A user has the possibility to login if he enters his username and password correctly.

```php
$usernameinput = $_POST["username"];
$passwordinput = $_POST["password"];



$querycurrentrealname = "SELECT realname FROM User WHERE username = '$usernameinput'";
$resultcurrentrealname = $conn->query($querycurrentrealname);
$row = $resultcurrentrealname->fetch_assoc();
$current_realname = $row['realname'];

$querycurrentisadmin = "SELECT isadmin FROM User WHERE username = '$usernameinput'";
$resultcurrentisadmin = $conn->query($querycurrentisadmin);
$row = $resultcurrentisadmin->fetch_assoc();
$current_isadmin = $row['isadmin'];

$querycurrentpassword = "SELECT password FROM User WHERE username = '$usernameinput'";
$resultcurrentpassword = $conn->query($querycurrentpassword);
$row = $resultcurrentpassword->fetch_assoc();
$current_password = $row['password'];

if ($passwordinput==$current_password) {
    $_SESSION["username"] = $usernameinput;
    $_SESSION["isadmin"] = $current_isadmin;
    $_SESSION["realname"] = $current_realname;
    header("Location: main_page.php");
    exit();
} else {
    header("Location: index.php");
    exit();
}
```

Once the user is logged in, he will be in the main menu. There he can choose to start playing the game. The application will display Google Street View imagery, a map in the bottom right side of the screen, a home button for the user to press if they want to go back to the main menu, their real name and after they double click on a place on the map they will have the option to register the guess by pressing a button with the text “Guess now”. After making the guess, the user will receive feedback on their accuracy and the number of points awarded. Then they can click next round and play again. Here is the end product:



The application tracks their statistics, the number of rounds played, the total score of all the rounds they have played combined together, the average score and the best score they had on the round. It is worth mentioning again here, like we did in the fourth chapter, the maximum amount of points will be 1000.

In terms of administrative features, and admin will have the ability to access a web page through the interface from which he can delete a user or promote a user to the admin role, if they confirm the action. Here is the implementation of the action of promoting a user to an admin role:

```php
$usertopromote = $_GET["username"];

$query = "UPDATE User SET isadmin = 1 WHERE username = ?";
$stmt = $conn->prepare($query);

if ($stmt) {
    // Bind the parameters and execute the statement
    $stmt->bind_param("s", $usertopromote);
    $stmt->execute();

    // Check if the query was successful
    if ($stmt->affected_rows > 0) {
        // User promoted successfully
        header("Location: manage_users.php");
        exit();
    } else {
        // Error occurred while promoting user
        header("Location: manage_users.php?error=promote_failed");
        exit();
    }
} else {
    // Error occurred while preparing the statement
    header("Location: manage_users.php?error=promote_failed");
    exit();
}
```

6.2.2. Technical requirements

In terms of the programming languages that were used, the application was developed using PHP for server-side scripting and HTML, CSS and JavaScript for the client-side functionality. PHP was very useful for interacting with the database, managing the users` information, while JavaScript ensured a dynamic and interactive interface, but was mainly used to deal with the interactions of the user with the map and Google Street View imagery. HTML and CSS were used to improve the aesthetic quality of the web application.

In terms of data management, we have discussed our current methodology in the fifth chapter, that we will improve upon in future iterations of the application. To discuss it briefly we use MariaDB which we manage via phpMyAdmin. The structure of our only table in the database is the following:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	**username**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
3	**password**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
4	**realname**	varchar(255)	utf8mb4_general_ci		Yes	USER			Change Drop More
5	**isadmin**	bit(1)			Yes	0			Change Drop More
6	**roundsplayed**	int(11)			Yes	0			Change Drop More
7	**bestscore**	int(11)			Yes	0			Change Drop More
8	**avgscore**	decimal(10,2)			Yes	0.00			Change Drop More
9	**totalscore**	int(11)			Yes	0			Change Drop More

We have to also discuss the third-party services that were integral in our process of developing the application. We integrated the Google Maps API for geolocation services and displaying the map and street view imagery. Furthermore, the documentation from the Google Maps developers were very helpful in managing to display the map and street view imagery and manipulate the characteristics during the display.

6.2.3. User requirements

Approaching the subject of user requirements, in our app there are two types of users, regular users and administrators. There is a field in the table user called isadmin, which determines whether a user is an admin or not, if the field value is 1, the user is an admin, and if it is 0, he is just a regular user. Administrators have additional capabilities compared to a regular user, they can access a page from where they can delete a regular user or promote them to an administrator role upon confirmation.

In terms of user experience, the interface is intuitive and user-friendly, allowing easy navigation and interaction with essence of the application, represented by attempts at geolocation guessing.

6.2.4. Design requirements

The application has a responsive, dynamic and interactive design, it follows modern web design principles, a clean dark mode layout and intuitive navigation. The visual elements of the

application, such as buttons, forms and menus are styled accordingly using CSS and they are well organized for general use with the help of HTML.

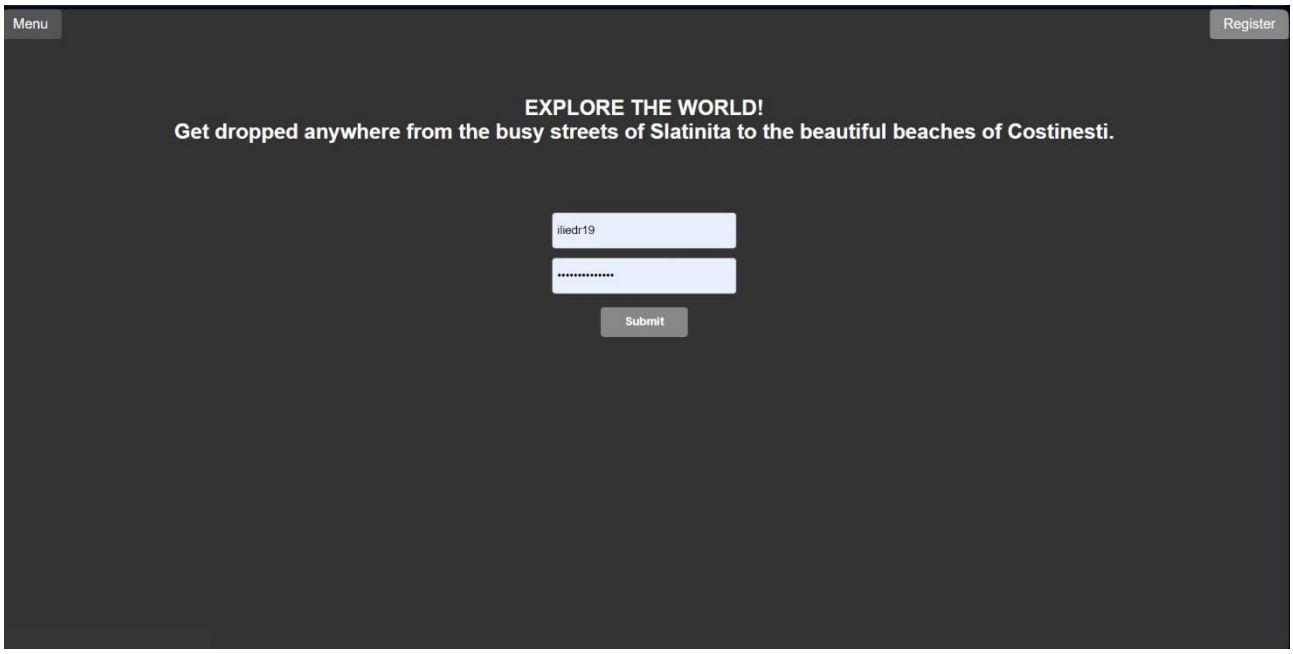
6.2.5. Data management requirements

As we have discussed previously on this matter, there are plenty of things to improve upon this that we will include in our Future work chapter. For now, the user account information and the game statistics are stored securely in the table user in the MariaDB database managed in phpMyAdmin. It can efficiently manage and retrieve relevant data during the usage of the app.

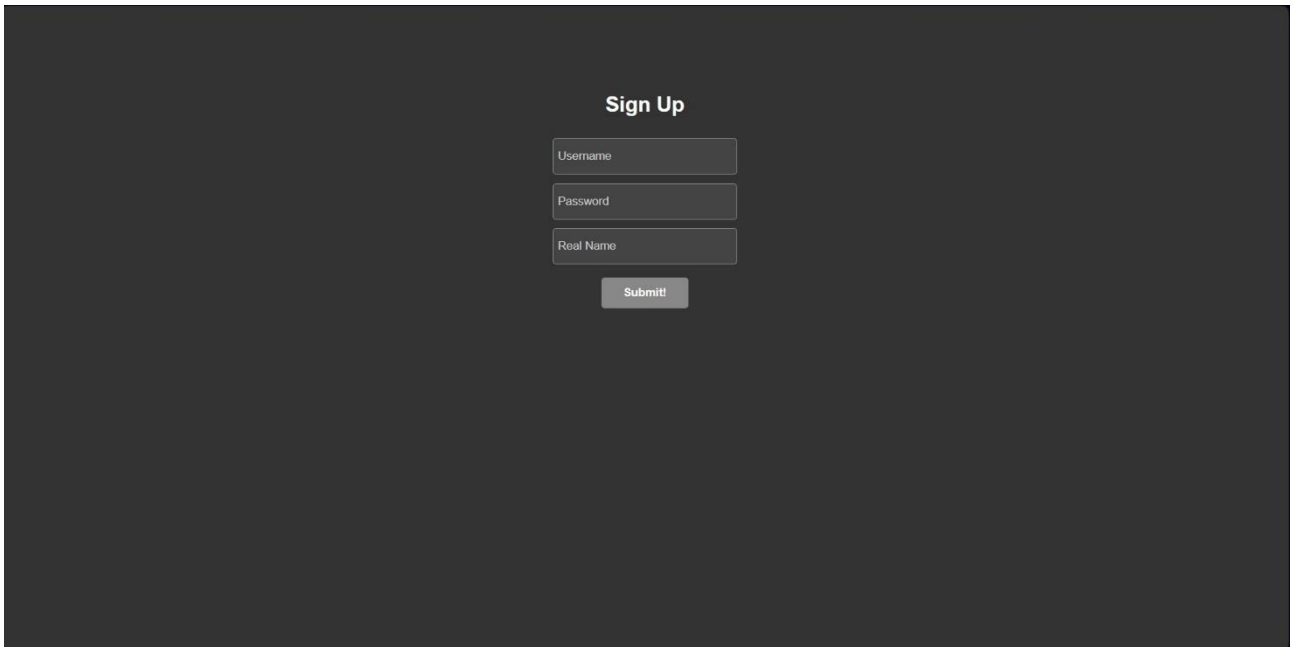
6.3. Design

In terms of design, the application “Locaisseur” will have a modern dark mode and the layout of the menus, forms and other visual elements are well put together to ensure a good user experience. We will show you the pages that a user will be able to see while navigating the application.

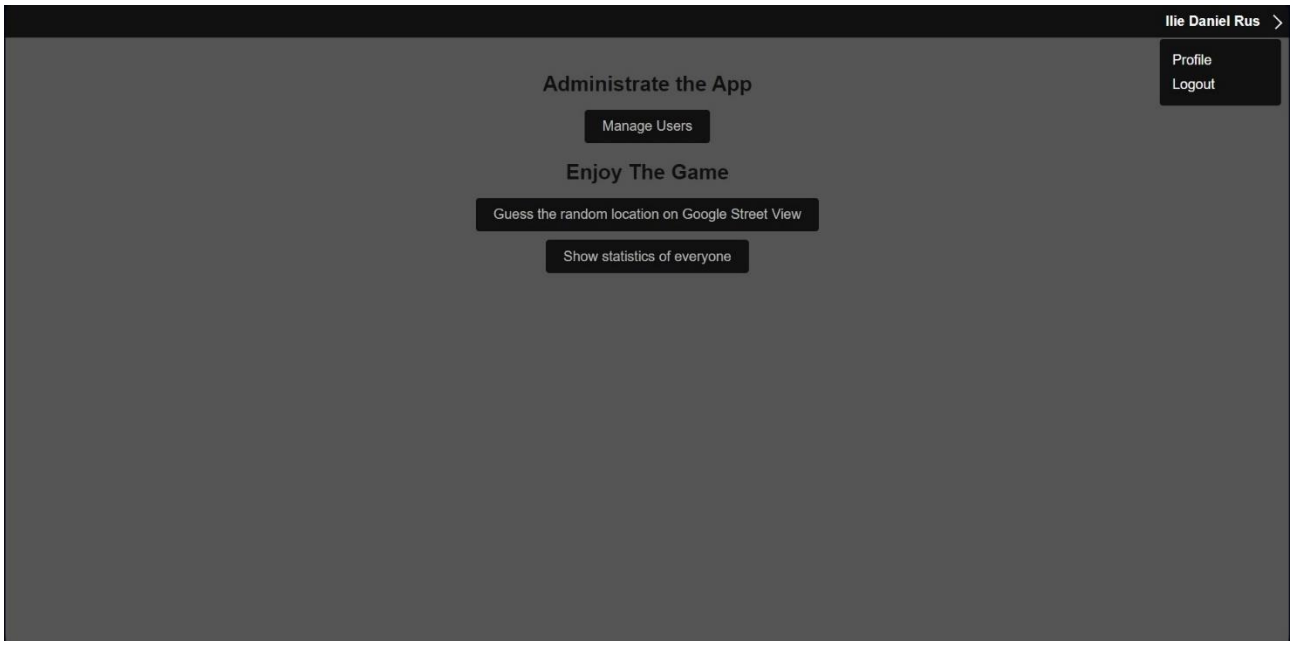
The first page after accessing the application:



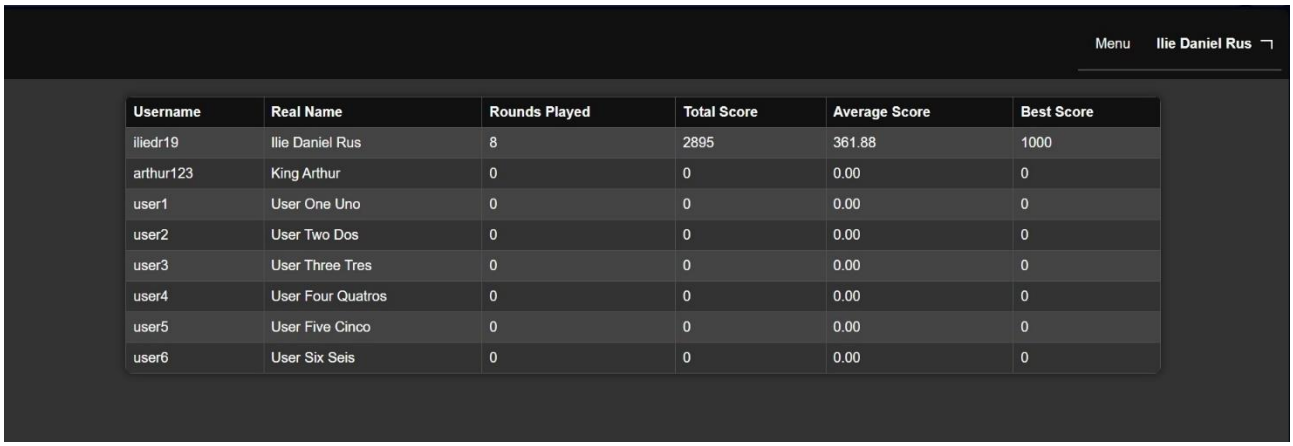
Register page:



Main menu:

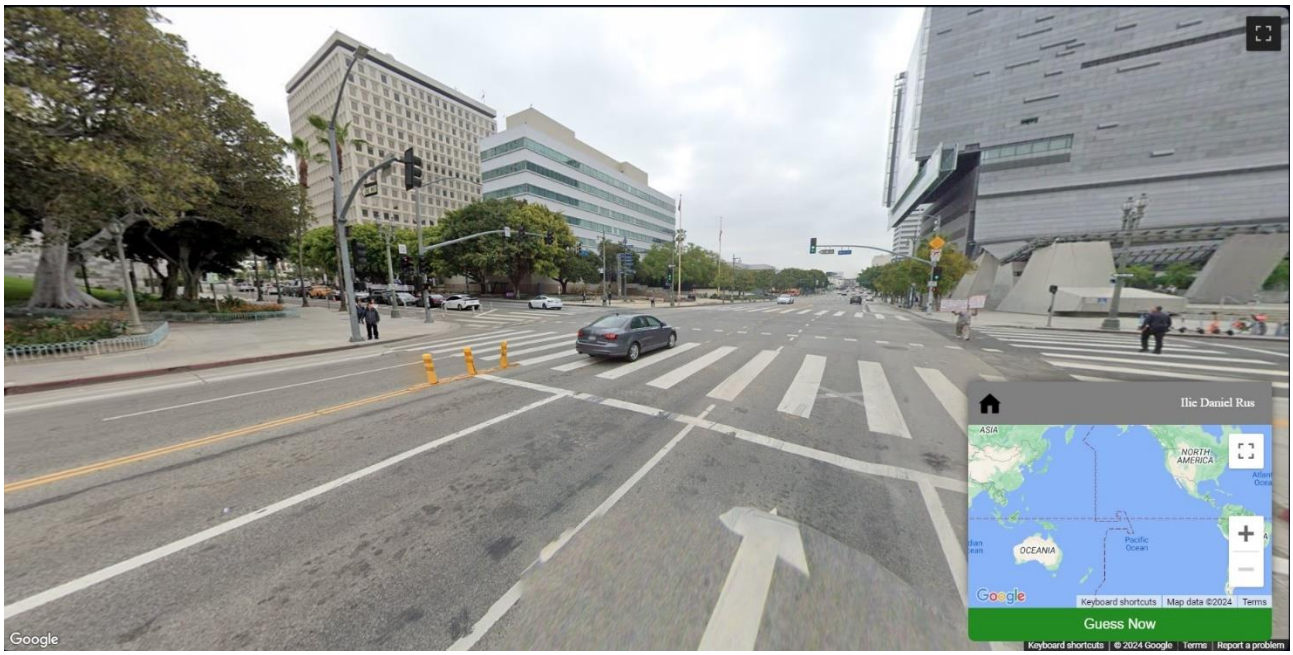


The statistics page:

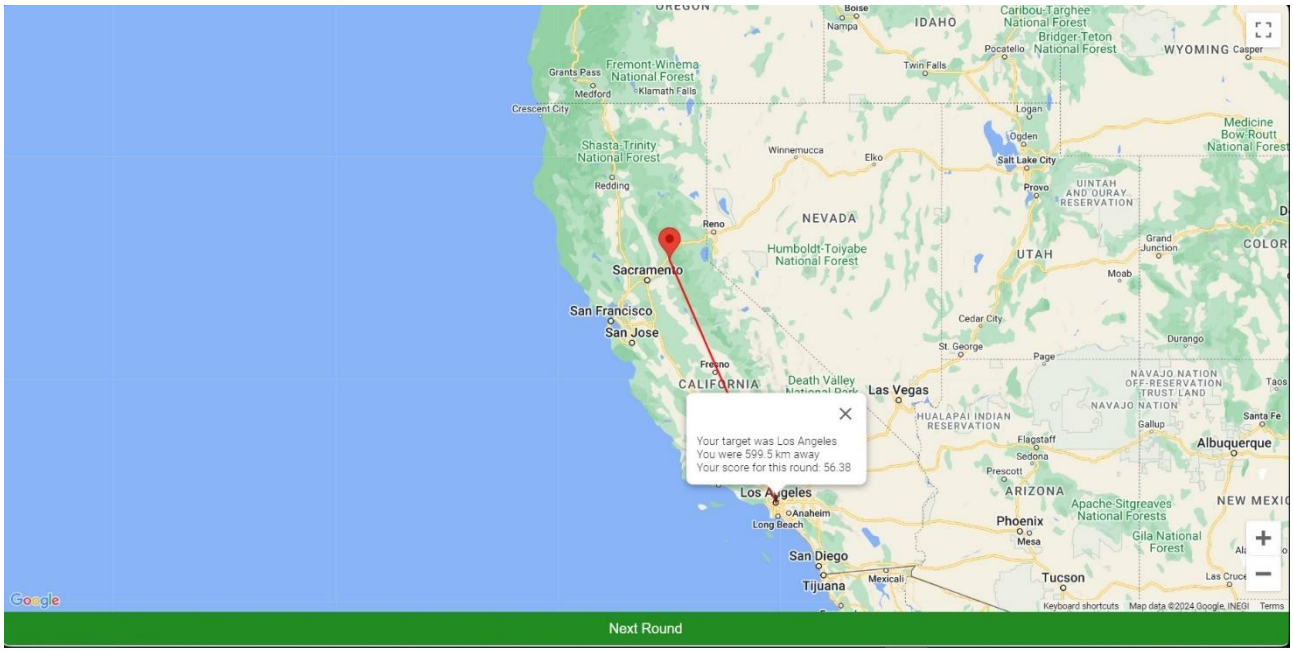


Username	Real Name	Rounds Played	Total Score	Average Score	Best Score
iliedr19	Ilie Daniel Rus	8	2895	361.88	1000
arthur123	King Arthur	0	0	0.00	0
user1	User One Uno	0	0	0.00	0
user2	User Two Dos	0	0	0.00	0
user3	User Three Tres	0	0	0.00	0
user4	User Four Quatros	0	0	0.00	0
user5	User Five Cinco	0	0	0.00	0
user6	User Six Seis	0	0	0.00	0

The gameplay page before user saves the guess:



The gameplay page after the user has confirmed his guess:



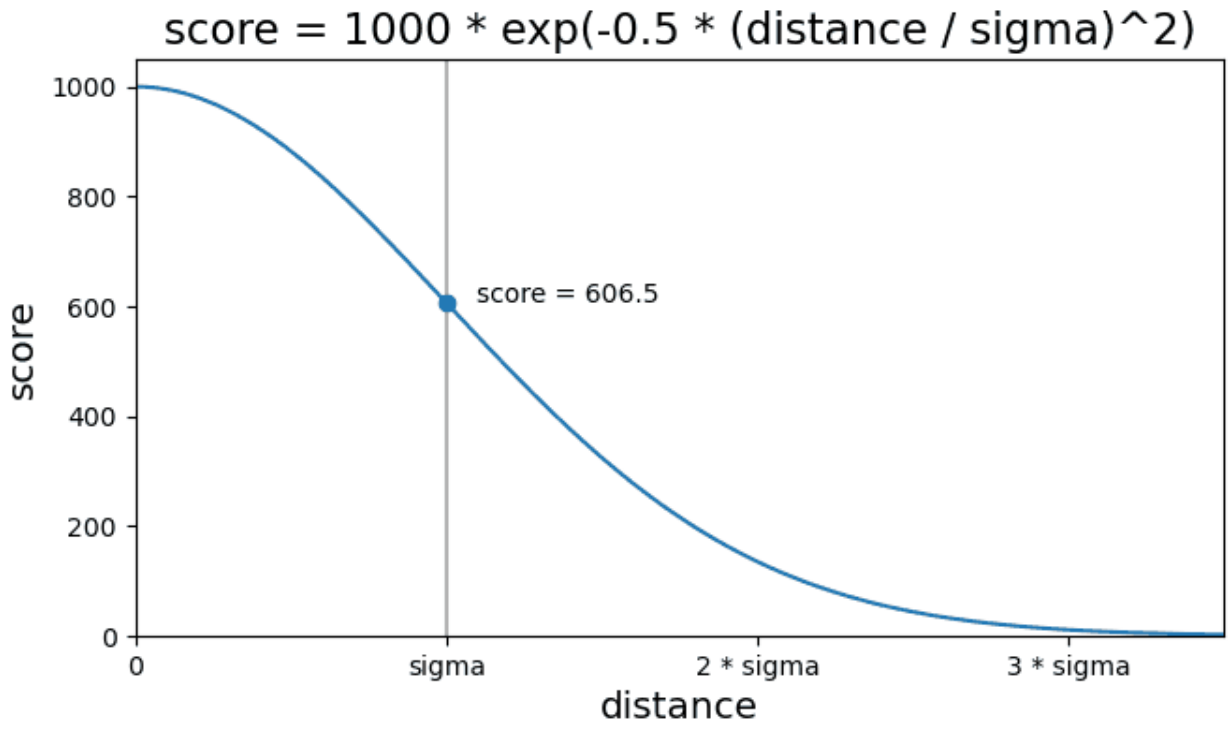
6.4. Implementation

In this subchapter we will briefly go through the main algorithms that have been used to create this project, namely calculating the distance between the point on which the user has double clicked with his cursor and later confirmed his guess using the button below and the target point, represented by the location of the Google Street View imagery that is displayed on the screen in that round. The other algorithm that we will look at is the point system, which is based on the distance between the two points we talked about earlier.

The Haversine formula calculates the shortest distance over the earth's surface, taking into account the curvature of the Earth. The process begins by retrieving the latitude and longitude coordinates of both the user's guess and the target location. These coordinates are then converted from degrees to radians. The differences in latitude and longitude are calculated and also converted to radians. The Haversine formula is applied to these differences, involving trigonometric functions to compute an intermediate value 'a'. This value is then used to find the angular distance in radians, which is multiplied by the Earth's radius (approximately 6371 kilometers) to yield the distance in kilometers. The Haversine formula:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

For scoring, the distance is obtained from the checkDistance function and used in an exponential decay formula to calculate the score. The score decreases rapidly as the distance increases, reflecting the accuracy of the guess. The formula normalizes the distance by dividing it by a constant sigma (250 in our case), squares the result, takes the negative half, and applies the exponential function to it. The final score is scaled between 0 and 1000, where a perfect guess (distance of 0) results in a score of 1000. The scoring formula and the graph of the value will look like this:



6.5. Testing

Testing plays a pivotal role in programming as it ensures the quality, functionality and most importantly, the security of the users while navigating and interacting with the application[19]. In our case, the testing will be driven towards the user authentication and the geolocation guessing functionalities.

In terms of testing user authentication, we need to check resistance to SQL injection[20] and test the usability of the interface, especially during the registering and logging in. It is important to give clear error messaging, as well.

Regarding the geolocation guessing functionalities testing, we need to verify gameplay mechanics, the feedback in terms of distance and scoring accuracy, the guess submissions and map navigation. Further down the line we will look into making sure the application works across different devices.

7. Results and conclusions

The current product is not final, there is room for scalability, but the foundations are very solid. We have developed a web application that can allow a user to register/login and play regular rounds of geolocation guessing, while his stats are being tracked and updated. The elements that are novel in the development of such an app in our domain are an easily navigable and minimalistic menu, as most of the fanbase that the most popular game in our branch, Geoguessr would want. Additionally, the game is free to play and will remain so, unless it will become extremely popular and the costs of maintaining the application will raise significantly. Also as a bonus, the application`s code is free source and can be found on Github.

Lastly, it is a great achievement to be able to combine the technological capabilities that I have acquired during these past three years studying computer science, my passion for geography, and implicitly for a game in this domain, resulting in a web application that can serve as an educational gaming app, a great tool to enhance problem solving skills and virtual exploration of our precious world.

8. Future work

As we have mentioned several times already in this thesis, any ambitious project, such as this will always room for improvement. I will mention some of the elements that can be added or improved upon in this chapter.

There is potential for growth in the aesthetic aspect of the app, while not making it result in a less user-friendly interface. We can add some dynamic displays of the earth in the menu backgrounds, to give the impression of a game, in order to attract more users. Also, the visual elements in the game, such as the home button or guessing button can be improved upon.

In the near future the application will be deployed, and it will result in an appeal to a wide and growing audience. That will happen following the improvement of the security within the application and thorough testing of the functionalities, mainly the login and register ones.

To make the game more interesting we can add more locations manually, or let the users create their own maps with hand-picked locations, that they can publicize in the application. An algorithm that will generate these locations will be a topic of importance to our application in the future.

Lastly, there are several modes and features in Geoguessr that are popular and worth looking into, as it will be able to attract more users to our application. One of them is no moving, panning or zooming mode, which more skilled users will enable in order to give themselves a harder task to guess where they are[21]. This consists of simply not allowing the user to move, pan or zoom the Google Street View imagery at all. Another mode is challenge mode, which generates a link for the user that plays a set of 5 rounds, which they can share with their friends, who will be able to get the exact same Google Street View imagery. However, one demanded feature that could set us apart, due to it not being implemented in any game similar to this, would be allowing the user to click on the location they have picked as a guess and see the exact Google Street View imagery from that location, or the closest available Google Street View imagery from that point. Such a feature could provide the user an idea on whether the guess they made was logical and they recognized a certain pattern that was similar in a similar area. For example, even though Canada and Norway are very far away from each other, even skilled users can be deceived into picking Canada when the Google Street View imagery has a location from Norway displayed, as both countries have yellow centre road lines, tall trees, scarcely populated areas all across the countries and similar houses design. A user of such an app would be interested in knowing whether they recognized such a pattern.

9. References

[1] https://www.geoguessr.com

[2] https://stackoverflow.com/questions/65351282/based-on-distance-away-from-a-coordinate-generate-score-lower-distance-away

[3] Larsen, C. H. P. (2022). Exploring Hedging Strategies for Multiple Players in GeoGuessr.

[4] Mehta, H., Kanani, P., & Lande, P. (2019). Google maps. International Journal of Computer Applications, 178(8), 41-46.

[5] Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., ... & Weaver, J. (2010). Google street view: Capturing the world at street level. Computer, 43(6), 32-38.

[6] Gibson, R., & Erle, S. (2006). Google maps hacks. " O'Reilly Media, Inc.".

[7] https://developers.google.com/maps/documentation

[8] Tungichbekovna, Y. L. (2023). THE IMPACT OF EDUCATIONAL GAMES ON ENHANCING ENGLISH LANGUAGE LEARNING ABILITIES. Научный Фокус, 1(7), 1086-1089.

[9] https://www.bloomberg.com/news/newsletters/2021-07-01/maplab-the-pandemic-era-appeal-of-geoguessr

[10] https://shotkit.com/news/this-new-ai-geolocates-photos-with-breathtaking-speed-and-accuracy/

[11] Girgin, M. (2017). Use of Games in Education: GeoGuessr in Geography Course. International Technology and Education Journal, 1(1), 1-6.

[12] Bartholomew, D. (2012). Mariadb vs. mysql. Dostopano, 7(10), 2014.

[13] Hu, S., & Dai, T. (2013). Online map application development using Google Maps API, SQL database, and ASP .NET. International Journal of Information and Communication Technology Research, 3(3).

[14] Purvis, M., Sambells, J., & Turner, C. (2006). Beginning Google maps applications with PHP and Ajax. Michael Purvis, Jeffrey Sambells, and Cameron Turner.

[15] Dauni, P., Firdaus, M. D., Asfariani, R., Saputra, M. I. N., Hidayat, A. A., & Zulfikar, W. B. (2019, December). Implementation of Haversine formula for school location tracking. In Journal of Physics: Conference Series (Vol. 1402, No. 7, p. 077028). IOP Publishing.

[16] https://i.sstatic.net/cGCRm.png

[17] Ahmed, S., & Mahmood, Q. (2019, November). An authentication based scheme for applications using JSON web token. In 2019 22nd international multitopic conference (INMIC) (pp. 1-6). IEEE.

[18] Robbins, J. N. (2012). Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics. " O'Reilly Media, Inc.".

[19] Sandin, E. V., Yassin, N. M., & Mohamad, R. (2016). Comparative Evaluation of Automated Unit Testing Tool for PHP. International Journal of Software Engineering and Technology, 3(2), 7-11.

[20] Halfond, W. G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In Proceedings of the IEEE international symposium on secure software engineering (Vol. 1, pp. 13-15). Piscataway, NJ: IEEE.

[21] Browning, K. (2022). Siberia or Japan? Expert Google Maps Players Can Tell at a Glimpse. International New York Times, NA-NA.