BSc (Hons) Computer Science and Software Engineering

Computer Integrated Systems
Design Documentation of
Inter Galactica - War for Math

Lecturer: Dr. Marc Conrad
Lecturer-In-Charge: Dr. Yasas Jayaweera
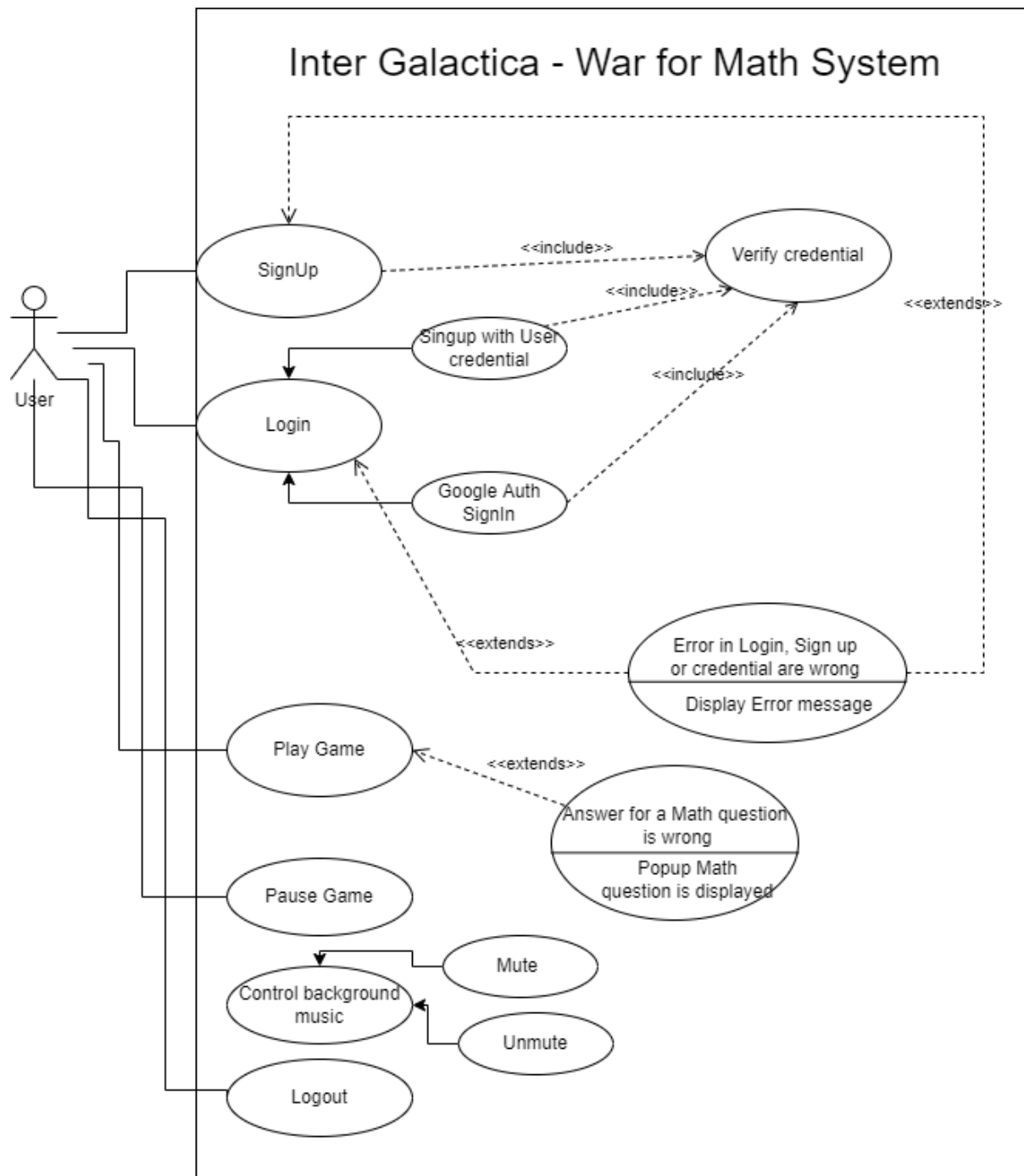
P.D. Buddhi Dhananjaya
2135868

Introduction

The "Inter-Galactica War for Math" is a retro style game created to interact with the users with the retro Game boy design, The Style and buttons are used within the game are created to looks like game boy. However, the functionalities of the game boy been changed. Since one of the assignment requirements was to make a game where users interact with the some form of mathematical game, I have utilized the HeartGame and developed it further by using various programing languages and concepts. Furthermore, to add a uniqueness to the game, I have created random math question generating API endpoint by using the random-math-question NPM library. Furthermore, Once the user land in the game play page, It will call MATHIMage api enspoint and take the HartGame image URL Question and the Answer, And Other api endpoint will be used for the RandomMathQuetion which is called if the game timed out or user didn't get the corect answer. The users have to find the value of the game under the specified time limit if the user is unable to calculate the answer for a question within the given time or user select the wrong answer then the user will be presented with a Error popup message and user will have to answer another question if the user get the answer correct he/she would gain 10 XP Points, however if the user is unable to get the answer for the random math question within the specified time or the answer is wrong then a couple of points will be deducted from the XP. Specially all the game settings are stored in a Json file by making it for the administrator to change easily. The aforementioned Json file contains the level settings for each level which contains the PlusScore, MinusScore and the XP points required to move to the next level. The Game also utilizes Ajax fetch request to update and get the details from the ClearDB MySQL Database.

The Game consist of 9 levels and the users can select the correct value for a heart, and gain points. The game is created with the OOP concepts and by utilizing the Singleton and Façade design patterns. Once the user login to the system it automatically calls to backend to get the user details form the database and store it in the Player class and the GameEngine class. GamEngine class acts as the main Singleton Object of the program. Since the game is only single player-based Singleton design pattern works quite well, however if there's a requirement to make the game Multi Player it can be easily done by using Sockets connections.
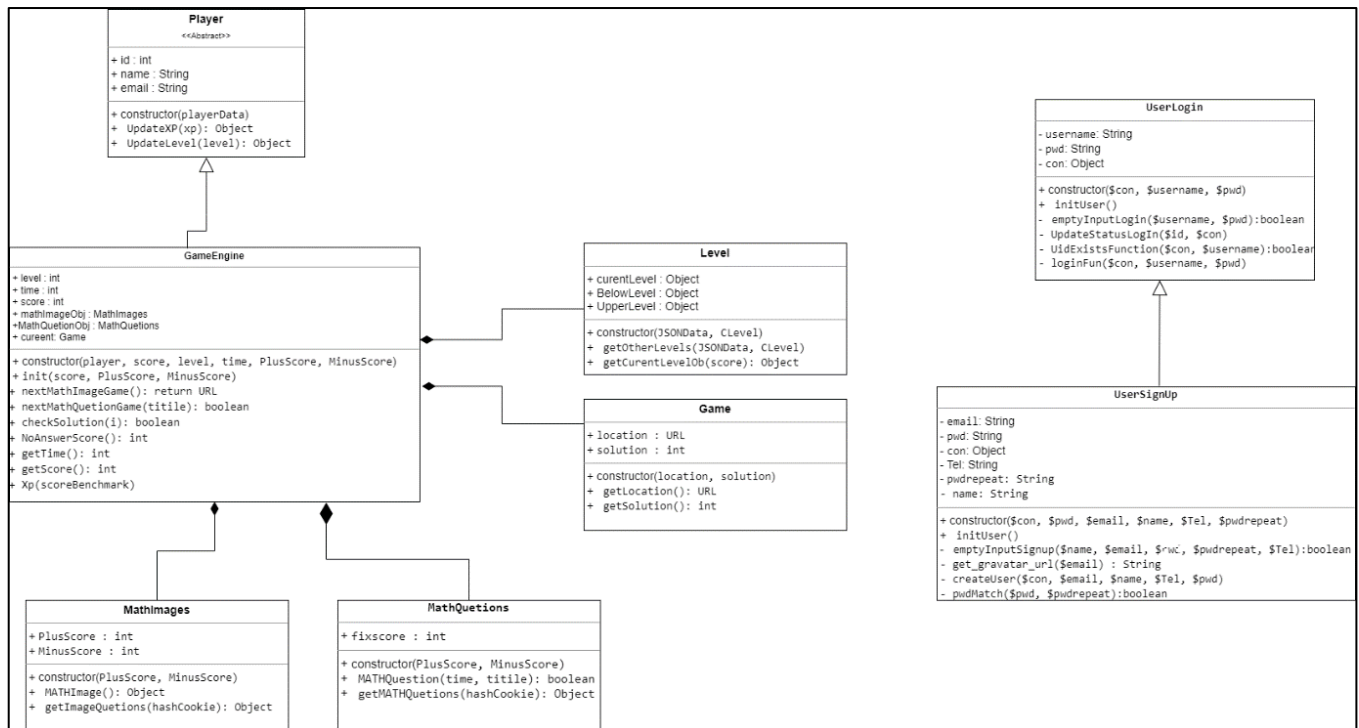
The entire system to work together it requires an active internet connection since both the Website (Game), Database (ClearDB MySQL) and Node js Server is hosted online on Heroku cloud pltform, Apart from the internel system compontents intenrt connection is also requirerd to connect to the externel systems.
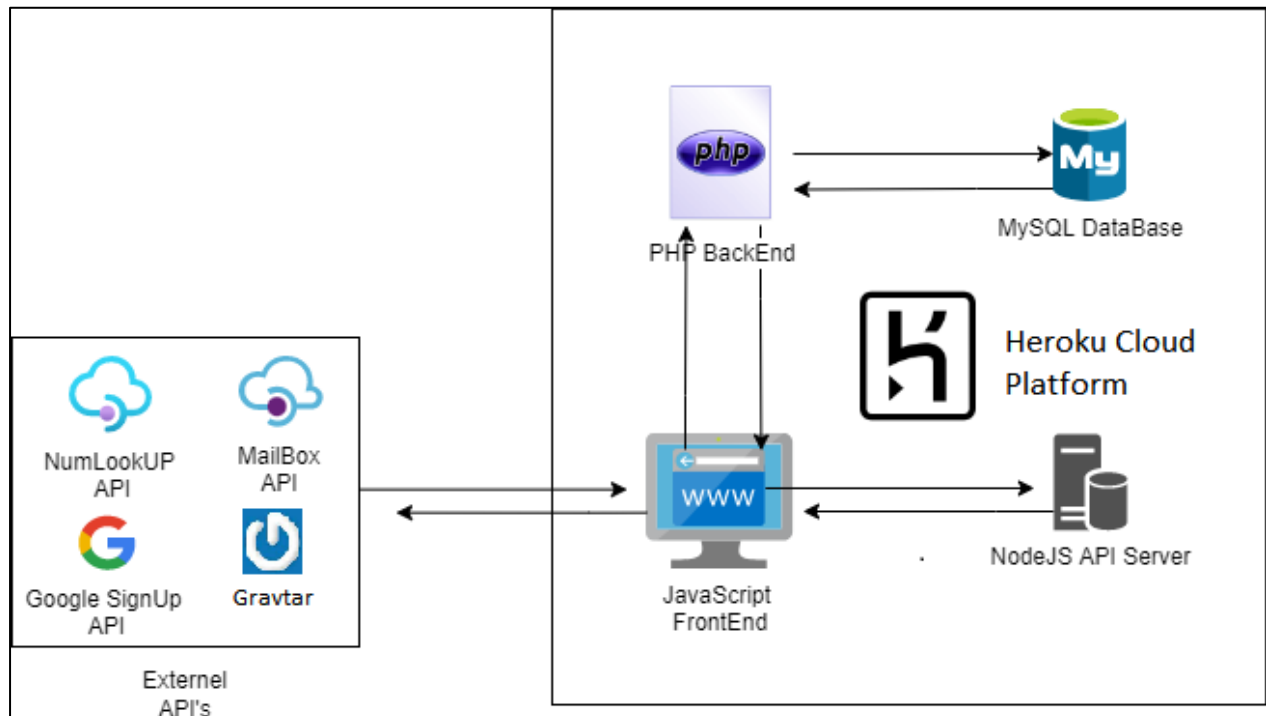
System Design and Architecture

Use case diagram

Use case diagram for Inter Galactica - War for Math System

Class Diagram

Player
<<Abstract>>

+ id : int
+ name : String
+ email : String

+ constructor(playerData)
+ UpdateXP(xp): Object
+ UpdateLevel(level): Object

GameEngine

+ level : int
+ time : int
+ score : int
+ mathImageObj : MathImages
+MathQuetionObj : MathQuetions
+ cureent: Game

+ constructor(player, score, level, time, PlusScore, MinusScore)
+ init(score, PlusScore, MinusScore)
+ nextMathImageGame(): return URL
+ nextMathQuetionGame(titile): boolean
+ checkSolution(i): boolean
+ NoAnswerScore(): int
+ getTime(): int
+ getScore(): int
+ Xp(scoreBenchmark)

Level

+ curentLevel : Object
+ BelowLevel : Object
+ UpperLevel : Object

+ constructor(JSONData, CLevel)
+ getOtherLevels(JSONData, CLevel)
+ getCurentLevelOb(score): Object

Game

+ location : URL
+ solution : int

+ constructor(location, solution)
+ getLocation(): URL
+ getSolution(): int

UserLogin

- username: String
- pwd: String
- con: Object

+ constructor($con, $username, $pwd)
+ initUser()
- emptyInputLogin($username, $pwd):boolean
- UpdateStatusLogIn($id, $con)
- UidExistsFunction($con, $username):boolean
- loginFun($con, $username, $pwd)

UserSignUp

- email: String
- pwd: String
- con: Object
- Tel: String
- pwdrepeat: String
- name: String

+ constructor($con, $pwd, $email, $name, $Tel, $pwdrepeat)
+ initUser()
- emptyInputSignup($name, $email, $rwc, $pwdrepeat, $Tel):boolean
- get_gravatar_url($email) : String
- createUser($con, $email, $name, $Tel, $pwd)
- pwdMatch($pwd, $pwdrepeat):boolean

MathImages

+ PlusScore : int
+ MinusScore : int

+ constructor(PlusScore, MinusScore)
+ MATHImage(): Object
+ getImageQuetions(hashCookie): Object

MathQuetions

+ fixscore : int

+ constructor(PlusScore, MinusScore)
+ MATHQuestion(time, titile): boolean
+ getMATHQuetions(hashCookie): Object

System architecture diagram

NumLookUP API

MailBox API

Google SignUp API

Gravtar

Externel API's

PHP BackEnd

MySQL DataBase

Heroku Cloud Platform

WWW

JavaScript FrontEnd

NodeJS API Server

Interface Design

Login Page

SignUp Page

Main Game Page

Test Cases

Test Case ID	Testing Scenario	Input	Testing Steps	Results Expected	Actual Results	Final Result
1	Login	Username: buddhi2002625@gmail.com Password: Ebf2dCEqVAVa6q2	1. Validate both fields for null values and If the values are not null 2. SQL PDO is run to check whether the field contains any unsafe operational codes, 3. The database returns the Username and Password if the user already exists.	Direct to the MainGame.php Under the hood Session should create along with 1 JavaScript cookie	Direct to the MainGame.php And a Sessoin and a Cookie was created	Pass
2	Login	Username: Jack@gmaisdl.com Password: Jack123hbsdfjkh	1. Validate both fields for null values and If the values are not null 2. SQL PDO is run to check whether the field contains any unsafe operational codes, 3. the database does not return anything since there are no existing user.	Dialog Box with username password is wrong	Dialog Box with username password is wrong	Pass
3	SignUp	Name: Buddhi Dhananjaya Email address: buddhi.jaya@gmail.com Phone number: +94 234 2312 22 Password: GHT123123 Confirm Password: GHT123123	1. Clicks the submit button 2. Checks whether all the fields are not null. 3. Validate the email by using (api.trumail.io) API service and the valid response is received. 4. Then validate the password, the password must contain 8 - 10 characters. 5. The validated password is checked with the confirm password to	Dialog Box with the message "Now you can Play and Learn Maths, and Enjoy Maths! Please Sign-In!" is shown and ask the user to login.	Dialog Box with the message "Now you can Play and Learn Maths, and Enjoy Maths! Please Sign-In!" is shown and ask the user to login.	Pass

			5. confirm that both the password are the same. 6. Validate the Phone number by using numlookupapi.com API service and the valid response is received. 7. Then the a SQL PDO is ran with the user entered email to check whether there's another entry already exist, No two entries found.			
4	SignUp	Name: Buddhi Dhananjaya Email address: Phone number: +94 234 2312 22 Password: GHT123123 Confirm Password: 123123	1. Clicks Submit button 2. Checks whether all the fields are not null.	Form submission is terminated and Tooltip is shown with the message directing that the email field is empty.	Form submission is terminated and Tooltip is shown with the message directing that the email field is empty.	Pass
4	SignUp	Name: Buddhi Dhananjaya Email address: Buddhi.jaya@gmail.com Phone number: +94 234 2312 22 Password: GHT123123 Confirm Password: GHT123GHT	1. Clicks Submit button 2. Checks whether all the fields are not null. 3. Validate the email by using (api.trumail.io) API service and the valid response is received. 4. Then validate the password, the password must contain 8 -	Form submission is terminated and Dialog box is shown with the message "Your passowrds don't match!, Please try again later!".	Form submission is terminated and Dialog box is shown with the message "Your passowrds don't match!, Please try again later!".	Pass

			10 characters. 5. The validated password is checked with the confirm password to confirm that both the password are the same.			
5	SignUp	Name: Buddhi Dhananjaya Email address: tempmail@g434l.com Phone number: +94 234 2312 22 Password: GHT123123 Confirm Password: GHT123GHT	1. Clicks Submit button 2. Checks whether all the fields are not null. 3. Validate the email by using (api.trumail.io) API service and the valid response is received. the same.	Form submission is terminated and Dialog box is shown with the message "The email you entered is not a valid email! You need to have a Valid Email Address!".	Form Form submission is terminated and Dialog box is shown with the message "The email you entered is not a valid email! You need to have a Valid Email Address!".	Pass

External System/API Integration
SignUp using Google OAuth cadential API and PHP SDK
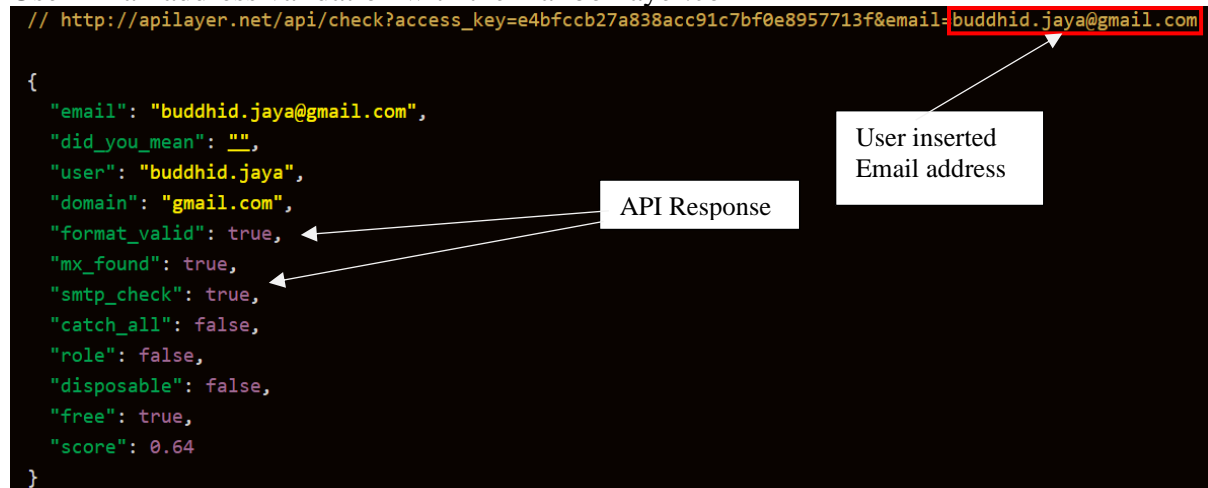

User Mobile number validation with the numlookupapi.com

```
// https://api.numlookupapi.com/v1/validate/+94704323860?apikey=pLWWIdzxTymJ9PSs7WQfg3KDOqMFv4EMgI7MLv8O

{
  "valid": true,
  "number": "94704323860",
  "local_format": "0704323860",
  "international_format": "+94704323860",
  "country_prefix": "+94",
  "country_code": "LK",
  "country_name": "Sri Lanka (Democratic Socialist Republic of)",
  "location": "",
  "carrier": "Mobitel (Pvt) Ltd (SLT-Mobitel)",
  "line_type": "mobile"
}
```

User inserted Phone number

API Response

User Email address validation with the mailboxlayer.com

```
// http://apilayer.net/api/check?access_key=e4bfccb27a838acc91c7bf0e8957713f&email=buddhid.jaya@gmail.com

{
  "email": "buddhid.jaya@gmail.com",
  "did_you_mean": "",
  "user": "buddhid.jaya",
  "domain": "gmail.com",
  "format_valid": true,
  "mx_found": true,
  "smtp_check": true,
  "catch_all": false,
  "role": false,
  "disposable": false,
  "free": true,
  "score": 0.64
}
```

Gravatar implantation for user profiles

Gravatar external service is used to get the user's profile picture at the time of signup, If the user already have entered a profile picture in their google or yahoo or for another email account then the gravatar would extract it and will produced a unique URL. That URL is then inserted in to the database within the user profile. This isn't a API service but a functional method implemented by the Gravatar group(gravatar.com, 2022).

Interesting Features

1. Extra level of Security to ensure the integrity of the Questions and Answers -- API call's Encryption and Decryption, Hence the user's won't be able to view the API request and Response since they are encrypted with the User Cookie details, base 64 key is used to encrypt & decrypt the API request-s and its data.

2. Easy game settings configuration -- Game Level settings configuration using a JSON file within the Game main directory. This will help the game developer to easily manipulate the game settings.

3. Improved cheat detection: Game Pauses and if the user tries to lookup an answer for a math question by opening a separate tab in the browser the game will continue to play

4. .htaccess file -- To disable directory browsing. Users can't browse through the different packages in the website and visit files stored in them.

```
Inter-Galactica > ⚙ .htaccess
   1    Options All -Indexes
   2    Options +MultiViews
   3
   4    ErrorDocument 500 "Sorry Folks.! The Inter-Galactica Server Is Down!"
```

5. Google Translate is implemented for Sinhalese, Tamil, Korean, Japanese, Simple Chinese, French, German. Since the website default language is English, Google Translate CDN library will try to translate the website into one of the above-mentioned languages after the user select an language from the drop down list, Once the language is selected, it will create a cookie containing the user's language of

preference. Moreover, from that point forward for 3 months user can use that selected language as the default language to aces the website.

6. Source controlling with GitHub to make the game workflow easier.

7. The artefact is hosted on Heroku SAAS platform, and can be accessed from: https://inter-galactica.herokuapp.com/

References

GitHub. 2022. *GitHub - tokyojack/Random-Math-Question: A function that generates a math question depending on the values.*. [online] Available at: <https://github.com/tokyojack/random-math-question> [Accessed 4 May 2022].

W3schools.com. 2022. *W3Schools online HTML editor*. [online] Available at: <https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_progressbar_labels_js> [Accessed 4 May 2022].

GitHub. 2022. *GitHub - googleapis/google-api-php-client: A PHP client library for accessing Google APIs*. [online] Available at: <https://github.com/googleapis/google-api-php-client/> [Accessed 4 May 2022].

En.gravatar.com. 2022. *Gravatar - Globally Recognized Avatars*. [online] Available at: <https://en.gravatar.com/> [Accessed 4 May 2022].

jQuery, H. and Bhaskaran, S., 2022. *How can I display a tooltip message on hover using jQuery?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/1333546/how-can-i-display-a-tooltip-message-on-hover-using-jquery> [Accessed 4 May 2022].

W3schools.com. 2022. *How To Google Translate*. [online] Available at: <https://www.w3schools.com/howto/howto_google_translate.asp> [Accessed 4 May 2022].