UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
(Final project paper)

Razvoj informacijskega sistema za iskanje soigralcev za različne športe

(Development of an information system for finding teammates for various sports)

Ime in priimek: Filip Todosovski
Študijski program: Računalništvo in informatika
Mentor: doc. dr. Vida Groznik

Koper, september 2022

Ključna dokumentacijska informacija

Ime in PRIIMEK: Filip TODOSOVSKI

Naslov zaključne naloge: Razvoj informacijskega sistema za iskanje soigralcev za različne športe

Kraj: Koper

Leto: 2022

Število listov: 32 Število slik: 11 Število referenc: 9

Mentor: doc. dr. Vida Groznik

Ključne besede: informacijski sistem, šport, PHP, CodeIgniter

Izvleček:
V diplomskem delu predstavljamo razvoj informacijskega sistema, ki bo pomagal povezati ljudi, ki se želijo ukvarjati s športom, a nimajo dovolj soigralcev. Za rešitev tega problema smo implementirali informacijski sistem, ki bo prikazoval objave igralcev, ki iščejo soigralca, ter vse ključne informacije o igri. Uporabniki imajo možnost iskanja po objavah z določeno ključno besedo. Poleg možnosti iskanja lahko uporabnik naredi tudi svojo objavo, kjer lahko navede vse podatke o igri. Povezovanje uporabnikov je možno preko klepeta, ki je na voljo ob objavi. Pri razvoju informacijskega sistema smo posebno pozornost namenili podrobnemu razvoju zalednega dela z uporabo programskega jezika PHP in ogrodja CodeIgniter, ter odjemalski del z uporabo HTML, CSS, Bootstrap in z uporabo baze podatkov MySQL.

Key words documentation

Name and SURNAME: Filip TODOSOVSKI

Title of final project paper: Development of an information system for finding teammates for various sports

Place: Koper

Year: 2022

Number of pages: 32 Number of figures: 11 Number of references: 9

Mentor: Assist. Prof. Vida Groznik, PhD

Keywords: information system, PHP, CodeIgniter, Sports

Abstract:
In this thesis we present the development of an information system that will help connect people who would like to play sports but do not have enough teammates. Therefore, in order to solve this problem, we have implemented an information system that will show posts from players who are looking for a teammate including all the key information about the game. Users have an option to search through the posts using a specific keyword. In addition to the search option, the users can also create their own post in which they can specify all the information about the game. Connecting users is possible using the chat next to the post. For the development of this information system, special attention is given both to the detailed development of the back-end part (using the PHP programming language and the CodeIgniter framework) as well as to the client-side development (using: HTML, CSS, Bootstrap and the MySQL database.)

Acknowledgement

Here I express my gratitude to everyone involved in my final project paper. I would like to thank my mentor Assist. Prof. Vida Groznik, PhD who has been of the most help during working on my thesis and was always available to provide me with guidance and I would also like to thank my family and friends who have always been there for me.

Contents

List of Figures

List of Abbreviations

MVC Model View Controller
JS JavaScript
API Application Programming Interface
URL Uniform Resource Locator

1 Introduction

Sports are fun, competitive and they do not feel like a "workout". From tennis to running, from climbing to surfing, we believe this is the best way to stay social and healthy in an increasingly remote and static world. What are sports without players or teams?

Most sports are either played in teams or they involve at least two players. It has happened to many of us that sometimes we have a desire to play a certain sport that requires more players. Finding players for a certain sport can be really hard. Not everyone has friends that play sports or they are just busy with their own activities and cannot get together. How do people who are alone and who want to play a sport that requires more players resolve this problem? On one hand, they can show up on the spot and try to find some people to play with. However, it may happen that there are already enough players and teams and there is no need for another player.

On the other hand, there might be some teams who are missing one or two more players to play the game. If the team members are not able to find an additional player, then the whole team cannot play the game. However, what if there was somebody nearby who would be a perfect match to join the team?

Finding players and connecting with them is especially hard when you are "new" in town. When you move to a new town, or maybe a new country you do not know where to find players for the sport you are interested in. Furthermore, maybe there are other people nearby who also want to play but you cannot find them. That is why it would be much better if you could just use the internet to find more information about the sports that are played nearby. Therefore, now there is an opportunity of having a system that could help players find the right team for their sport and help teams with their lack of players. Moreover, this system could connect players within the same town who are interested in playing sport but do not have teammates. It will allow them to communicate with each other to arrange the right place and time for the game. Sport Finder is a free web application that can be used by everyone with internet access. Non-registered users are only able to access the web application and create an account. Registered users have access to all application functionalities such as: viewing posts, creating posts, modifying posts, communicating with other users, and rating.

This thesis is structured as follows. Section 2 presents some existing systems and their functionalities. Section 3 presents the functional and non-functional requirements. Section 4 presents the implementation and the technologies that are used. Section 5 provides the conclusion of the thesis.

2 Related work

Firstly, we will take a look at some of the existing systems: Sportifiq, Semaforum and social media where you can connect with people.

2.1 Sportifiq.com

Sportifiq[1] is a web application that allows users and teams to make a tennis court reservation. Sportfiq is based in Slovenia and it offers only the tennis courts in Slovenia. We have tried using this application. However, we were unable to create an account. No reasons why we could not do it were provided. We were unable to use most of the functionalities that the registered users had. As a result, we can conclude that the web application has not been maintained and that there have not been any new updates.

Sportifiq can be upgraded to be an excellent sport application that most of the sporty people will use if they implement more functionalities. First of all, most popular sport courts should be listed, which will attract more potential users. Furthermore, more user-friendly search/find function should be implemented, which will allow better access for users. What is more, some kind of a rating scale or a review section could also be implemented, which would be more useful to future customers.

The user-interface needs upgrade as well. There is a lot of unused space that can be somewhat confusing for new users. A recommender system can be very helpful in the sense that it will recommend sport courts to users based on their preferences.

2.2 Semaforum.si

Semaforum[2] is a web application that allows users to create and find activities based on their preferences. Although there are mainly sports activities, other activities such as: travelling, concerts, and dating can also be found. This application allows users to create an account so they can use more functionalities. As a non-registered user, you can only search through the posts, but you cannot connect with other people.

Semaforum has a clean user-interface that allows every user to have an excellent

[1] www.sportifiq.com
[2] www.semaforum.si

experience with their web application. It has a search bar where users can put all their preferences and where they can search for a specific category. By clicking on a post, we could find all the information about the location and about the person who had created the post. Nevertheless, adding a rating bar (or a user review) is the only thing that is missing. This can be improved in order to increase the trust of new users. Connection is made via message. Every post has a message window where every user can chat. As we were searching through the application, we found that there was an implemented recommender system which was supposed to make recommendations to every user based on their preferences or based on their searches. However, the recommendations which were found in this web application were very inaccurate. That is something that needs to be fixed so that users can get more accurate posts and they can have better connection with other users.

2.3 Social Media

Social Media, as a big influence in today's society, has its own positive and negative sides. Social media connects people around the world. Many social media sites such as: Facebook, Instagram, and Twitter have a lot of fake accounts which spread fake news.

Social media is a very powerful tool when it comes to communicating with your friends or people around the world. There are a lot of pages and groups that are very helpful. When it comes to sport and sports activities, everything can be arranged through a couple of messages. The communication with your sport buddies can be done through messenger which also allows the option to have a messenger group with more people who have the same interests. Furthermore, people who are new in the city and do not have any friends who are playing sports can join the messenger groups or the Facebook groups. They can search through the groups or they can simply add a post in which they can specify all the information and what they are exactly looking for.

Groups are useful but they can also be time- consuming as it can take some time to find the right people to connect with. There will be a lot of posts and you never know what is fake and what is real. The main disadvantages of groups are that we cannot search with a keyword through the posts. We need to search manually through all the posts. What is more, you do not have any kind of rating or user review, which can be very useful.

3 Our Solution

3.1 Functional requirements

1. The system should allow its users to create, edit and delete posts in which they can specify all of the necessary information such as: sport name, location and time, and number of missing players.

2. The post created by the registered users must contain: a title, a description and the date when the post was created, as well as the username of the person who created it.

3. Users should also have a way of navigating through the available posts with the help of a search function. The search function is performed via a search bar. Users write keywords in the search bar and the search begins. They can choose if they need a player or a team, the location, the date, and the time. In this way, users have the option to narrow down the search results to the ones matching their needs.

4. The system can be viewed by anyone but it can only be used by registered users. Everyone will be allowed to create an account. Registered users have the ability to create, edit, and delete entries in the form of posts. They will need to fulfill personal information when creating an account such as: an email, a password or a username.

5. Once the user finds a post they are interested in, the system should provide a way to connect the user with that post. There will be a messenger window so they can send direct messages to the owner of the post.

6. The system should have a function that allows registered users to rate the users they have played with. The rating scale is from 1 to 5, and the displayed rating is the average of all the received ratings.

7. Since the information system has a report option, it also needs some accounts that will work on those reports. Moderator accounts will have the ability to check all reports and remove those users who are posting inappropriate comments. The

report also needs to have a brief description why it has been made. Moderators will have the information about the user that has been reported and the user that reports. Only moderators would have access to these reports.

3.2 Nonfunctional requirements

1. The system should be able to support at least 100 users visiting the page at the same time.
2. The system should be able to process at least 10 user requests per minute.
3. Backups will be made on a 30-day period.
4. Since the system has the option to manage personal information, means of data protection are also needed. Secure communication between users is also required.
5. The search option for a sports game should be available to everyone. However, only registered users can connect with other users.
6. The system will be used by anyone with a personal computer/smartphone with an internet connection.
7. The system should recognize two types of users: registered and unregistered users.
8. The system must work 24/7.
9. The system is a web application.

4 Implementation

The system is implemented as a web application. The server-side is made with the PHP programming language and the framework CodeIgniter that provide database connection libraries where all the information is stored. The client-side is made with the HTML and CSS, which are the base for web development

4.1 What is "server side" of application?

The back-end part is actually that what is not visible to the user and what is in the background and it represents the so-called server-side of a web application. The frontend part is visible to the user and it actually represents the overall experience and view of the user, which is why it is called the client's side of the application.

How does the back-end architecture work? The back-end of every application represents a combination of the database and the software which is written in a server-based programming language. This combination is later executed on a specific web server, cloud based servers, or a hybrid combination of both. This section communicates directly with the database through a specific API (Application Programming Interface). [1]. The workflow of the Backend architecture is shown on Figure 1.

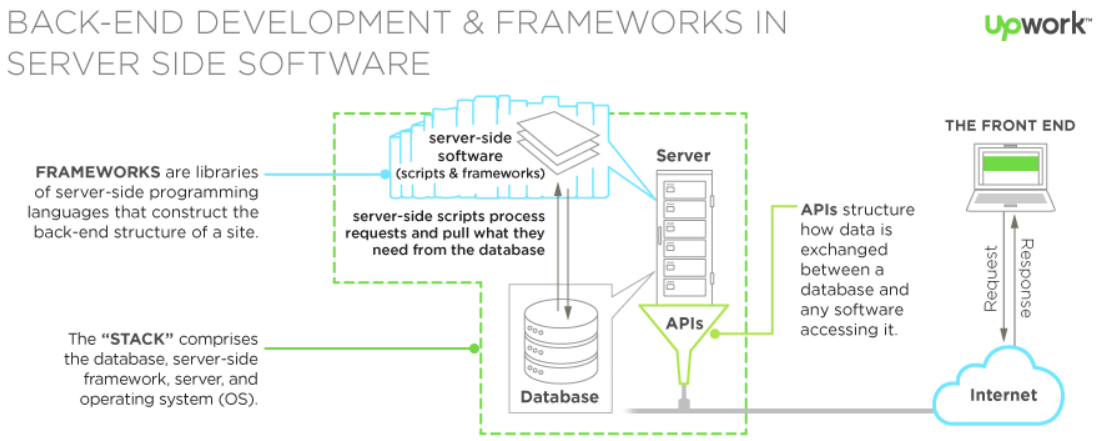

Figure 1: Backend architecture.
(Source: https://www.business2community.com/brandviews/upwork/beginners-guide-back-end-development-01855622)

4.1.1 PHP

PHP is a server-side scripting language designed specifically for web development to produce dynamic web pages. It was created by the Software Developer Rasmus Lerdorfin 1995. [2] The PHP code is embedded into the HTML source document and executed by a web server with a PHP processor module, which generates the web page document.PHP is compatible with a variety of databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. The client will never see the code generation part of the HTML content of a web page because the PHP code is run "server-side".

The client opens the browser and requests the web page. Then the server checks .php file associated with the request and the file is sent to the interpreter which checks for requested data into the database. The web server receives the requested data in HTML output and it sends the web page back to the browser. The workflow of the PHP programming language is shown in Figure 2.

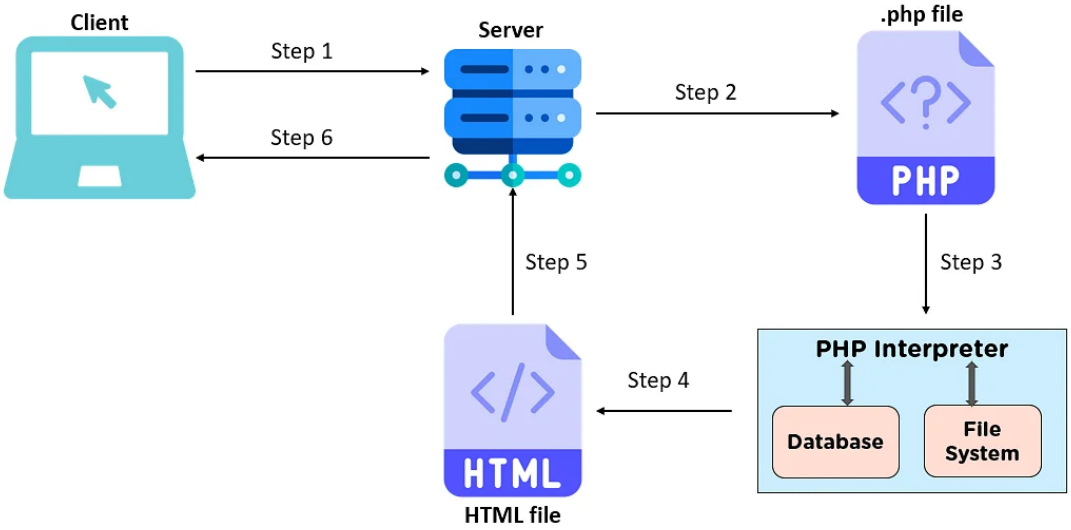

Figure 2: PHP workflow.
(Source: https://www.simplilearn.com/tutorials/php-tutorial/what-is-php)

4.1.2 CodeIgniter Framework

We use frameworks in order to get a simplified code structure and save time while writing code. Frameworks provide a place to start and to reduce the amount of code required to create a website. [4]
CodeIgniter [3] is a PHP Model View Controller (MVC) framework used for developing web applications. CodeIgniter offers pre-built libraries, which allows connecting to the database and carrying out various activities such as: sending emails, uploading files, managing sessions, etc. The source code of the CodeIgniter is very small. Therefore, learning CodeIgniter's functionality is simple. In CodeIgniter, when a user requests a resource, the controller responds first. The controller understands the user request, and then it requests the necessary data.

This picture shows the workflow of the CodeIgniter if we want to retrieve information about the user with id=3. Firstly, the user makes a request from the browser, the controller or the part where our code is written receives the request and sends it to the model or the database where the user with id=3 is searched for. The model sends the retrieved information to the controller where it is formatted into an HTML file and it is sent back to the browser. The workflow of the CodeIgniter framework is shown in Figure 3.

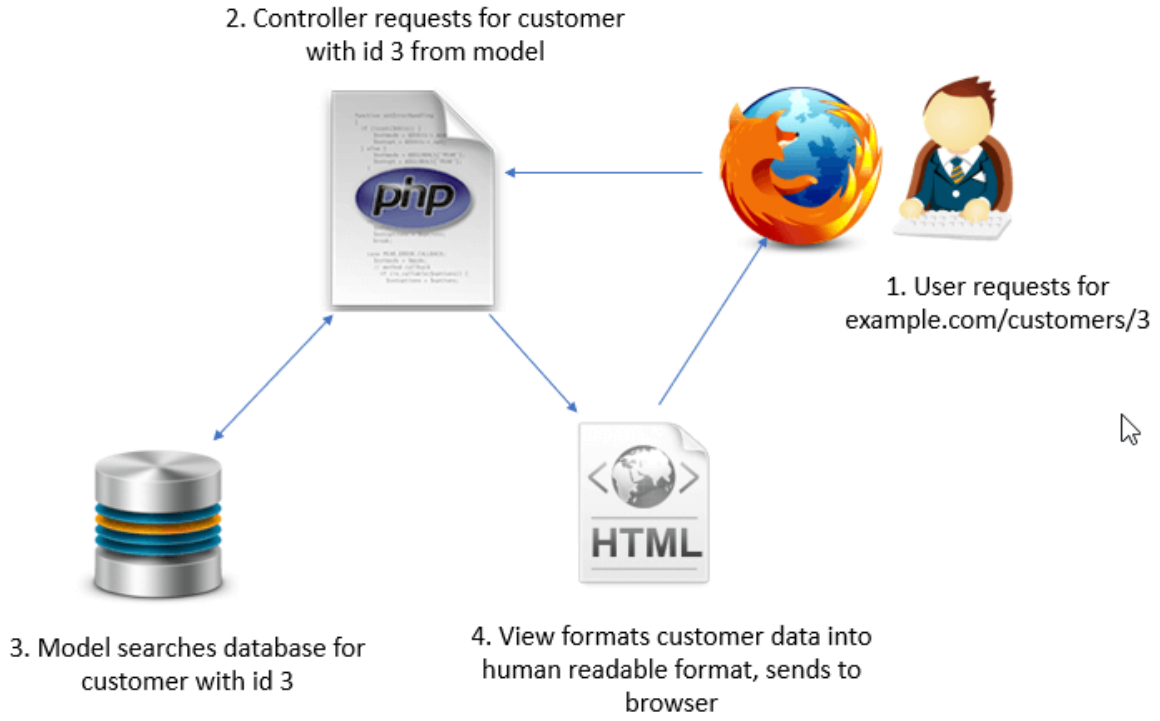

Figure 3: CodeIgniter workflow.
(Source: https://www.guru99.com/what-is-codeigniter.html)

4.2 What is "client side" of application?

In the previous chapter we briefly mentioned what the client-side of a web application actually meant. Successful businesses need a functional, clean and attractive website that will attract even more visitors and that will increase the number of their monthly users. The fact that the client-side of the application must be attractive, interesting, creative, and interactive is the main goal.

There are several rules when creating a successful client-side. First of all, we would like to point out that the front-end part is not related only to the program code. Yes, of course it is related to the code that the developer will write. However, the exact connection is that it is related to the code that has some interaction with the users. The front-end part gives the first impression of the application. When users get this impression, they also get an impression of the company itself. If the impression is positive, then the company is on track to get a new loyal user. Nevertheless, if the impression is negative, then the team working in the field of development should come up with a new strategy. [5]

Everything a user sees in the browser is a mixture of HTML, CSS and JavaScript. When the user runs the web page in the browser, the client-side scripts will need to processrequests from the user. If a call to the database is required, then requests are

sent to the server-side via JavaScript and AJAX. After that, the server-side processes the request, gets what is needed from the database and sends it back. The Frontend architecture is shown on Figure 4.

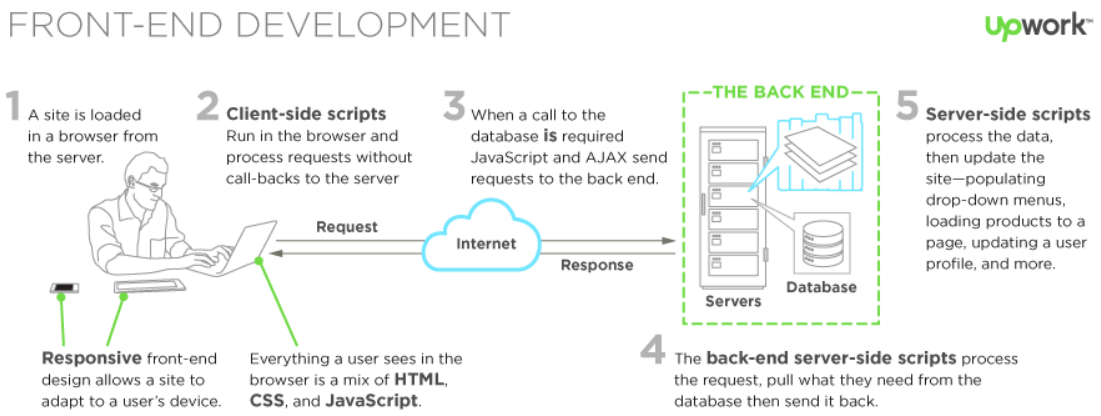

Figure 4: FrontEnd architecture.
(Source: https://www.business2community.com/brandviews/upwork/beginners-guide-front-end-development-01851787)

Users need to interact with a design that is user-friendly. The design should be quite simple and understandable so that the users are not confused. Users lose interest if they constantly encounter certain errors in applications. It is very likely that when the customer's trust is broken, then the business will suffer some losses. A clean and well-planned concept, impressive graphics and illustrations are functionalities that make the client-side a really important segment of an application.

4.2.1 HTML and CSS

It is necessary tolay a solid foundation for successful operation of the client-side of a web application. In most cases, this is based on the most widely used language in the client-side. The Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. HTML defines the structure and layout of a Web document by using a variety of tags and attributes which are interpreted by the web browser software to display the pages. There are hundreds of tags used to format and lay out the information in a Web page.

Although HTML is the basic structure of a web page, CSS is what gives a web page its unique style. CSS, or Cascading Style Sheets, allows dividing the content of

the document so that it includes the following elements: the font type, the display layout, the page elements, the colour palette etc. With this division, the page content is improved, getting a clear, interesting and striking visual change. CSS beautifies the basic structure of the content prepared through HTML: display colours, special interesting fonts and background images exist based on CSS. [6]

4.3 MySQL Database

Every web application that requires storing data needs to use database. Databases makes data management simple [9]. MySQL is an open source relational database management system. SQL stands for Structured Query Language which is a standardized language used to access databases. MySQL stores data in tables made up of rows and columns. Users can define, manipulate, control and query data by using SQL. MySQL is designed to handle a large number of workloads, from single machines to data warehouses or web services with many concurrent users [8]. It allows you to add custom functions developed using different programming languages such as PHP, C / C ++, Java, etc. [7]

4.4 Feasibility study

One possible solution to the problem is to develop a system that will allow users to find more players that play the same sport and connect with them. The system will be a HTML page where the users can select all the needed details for the best matching teammates. They can search for the ads that match their requirements. The system needs to satisfy all functional and non-functional requirements.

If we look at the solution from a technical standpoint, most of the functions require storing data and communication among users. The functionalities are not so difficult because the only thing they require is a dynamic website which will use a database. Constructing and maintaining the database will not be a problem. All the data in the database will be provided by the users themselves. As for the web server, it will be neither hard to obtain nor expensive to maintain. This is because the estimated traffic that the system will experience and the server will support is not high. When it comes to the server-side, all other functionalities that are likely to be required are not considered problematic as one should be able to use server-side scripting language (e.g PHP or JS (JavaScript)). The server-side scripting language will establish a connection between the web application and the database.

The nature of this system allows possible predatory behaviour exercised by its users. The best solution is to have a legal agreement in form of Terms and Conditions.

Every registered user will need to agree to a set of Terms and Conditions which will establish relationship between the information system and the user. In addition to the Terms and Conditions, Privacy Policy should also be defined since it is required by law in many countries. From the legal perspective, there are no restrictions that would render thissolution not feasible. The same goes for the cultural beliefs and norms, which implies that the solution is operationally feasible.

The information system does not require any paid software. It can be implemented by using free software and it can be hosted on the server provided by the faculty. This system is socially acceptable as it gives a very clear picture of people who love sports. Moreover, people can search, connect and group according to their needs. This information system will be of no economic burden to its users because it will be free of charge.

4.5 UML diagrams

Some of the UML (Unified Modeling Language) diagrams are presented in this subsection. UML diagrams are used to demonstrate the workflow of the system as well as the operation of all of the attributes that are inside the system. Each diagram has its own explanation which makes it easier for people that lack computer science experience.

4.5.1 UML Use Case diagram

The functionalities of the Sport Finder web application are shown in Figure 5. The user can create a post where they will need to specify some basic information about it and they will need to determine in which sport category it belongs to. After that, the user has the following options: to manage the existing ads, to edit some information or to delete the post. Furthermore, users can search through existing posts and they can choose which game they will join. By joining, the user will have the option to rate the creator of the post and to use the group chat.

The workflow of the system is shown on the UML use case diagram in Figure 5.

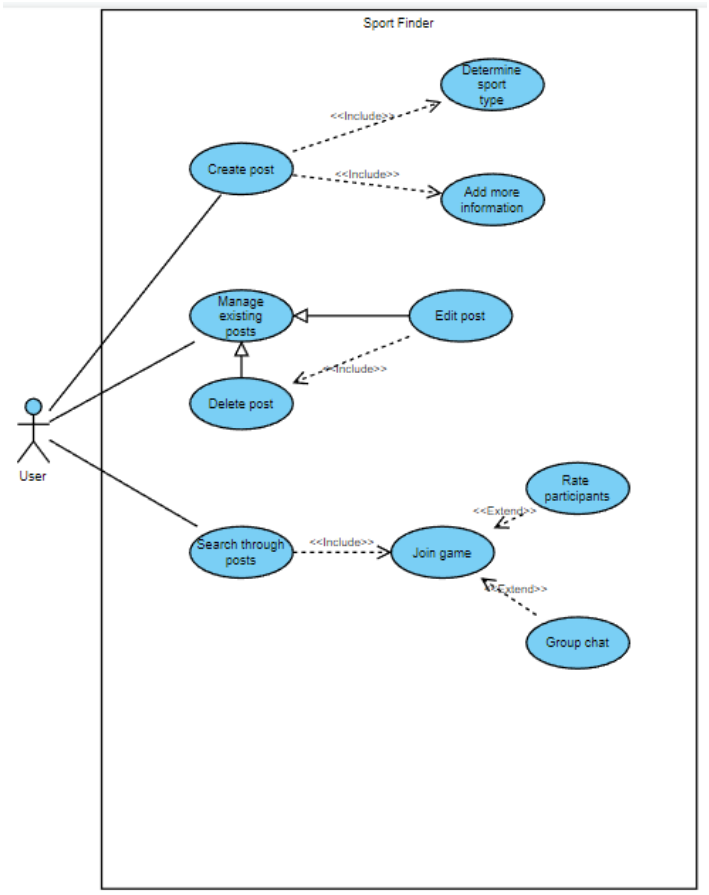

Figure 5: UML use case diagram representing the workflow of the system.

4.5.2 UML Class diagram

In The attributes and their operations in the system are shown in Figure 6. We have two types of users in our system: registered and non-registered users. The non registered user has only one option, which is to create an account so they can get access to all other functionalities in the system.

Registered users can create posts where they need to specify information that will be useful for the users they are searching for a game. As there are many sports, every post will belong to a sport category which will allow better filtering. There will be a messenger where users can join the game as well as a rating scale to rate the creator of the post.

The system connections are shown on the UML class diagram in Figure 6.

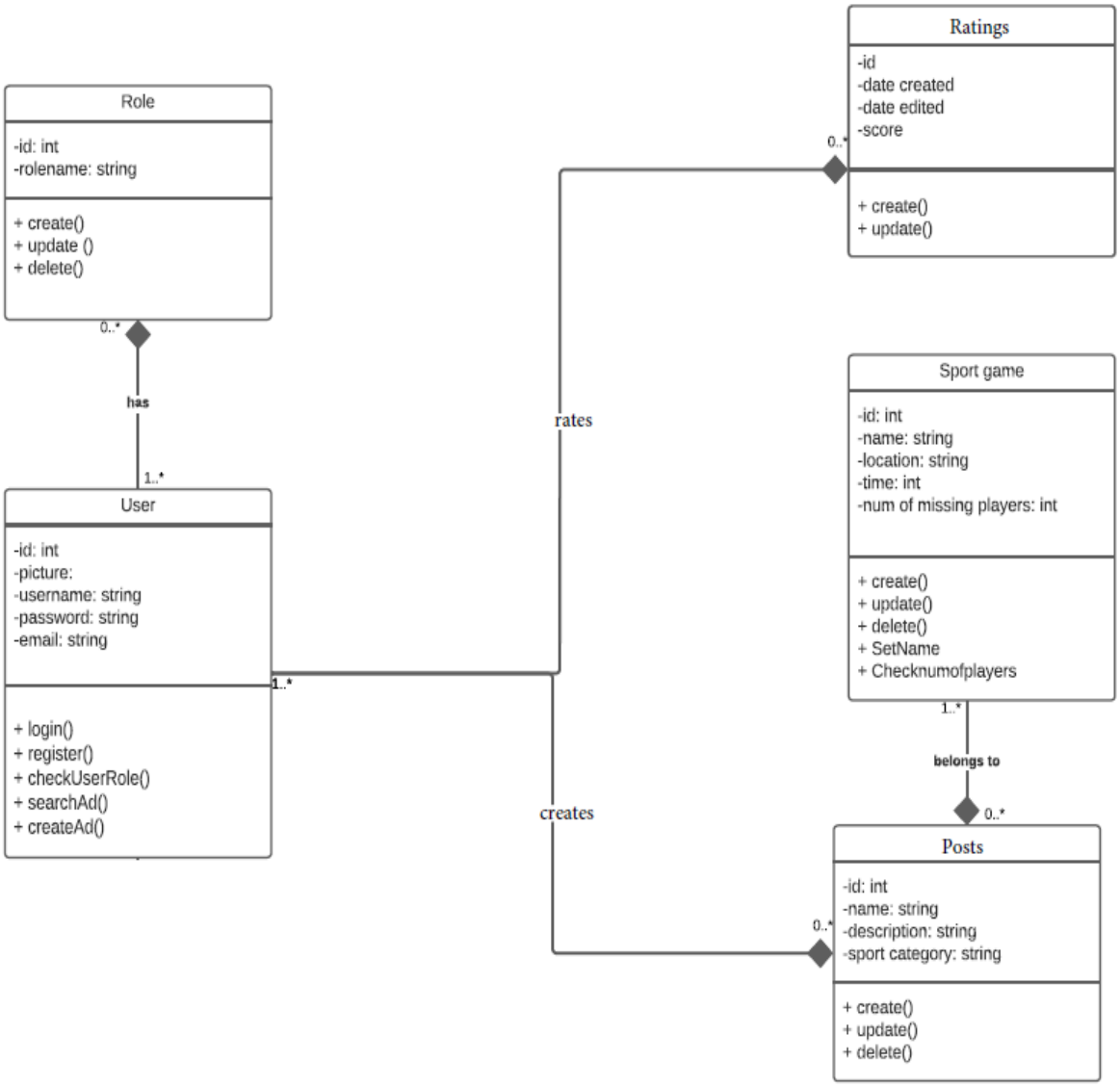

Figure 6: UML class diagram representing the attributes and their operations in system.

4.5.3 UML Sequence diagram

The process of creating posts can be seen in Figure 7. When the user is logged in, the application will verify the user's password in the database. If the password exists, then the user will get an option to create a post. If it does not exist, the user will get an error message. The user can create a post where they will need to fulfil all the blank spaces. If everything is fine, the post will be added to the database. If something is wrong, the system will throw an error. Finally, it is shown if the post is fine on the main page.

The process of creating posts is shown on the UML class diagram in Figure 7.

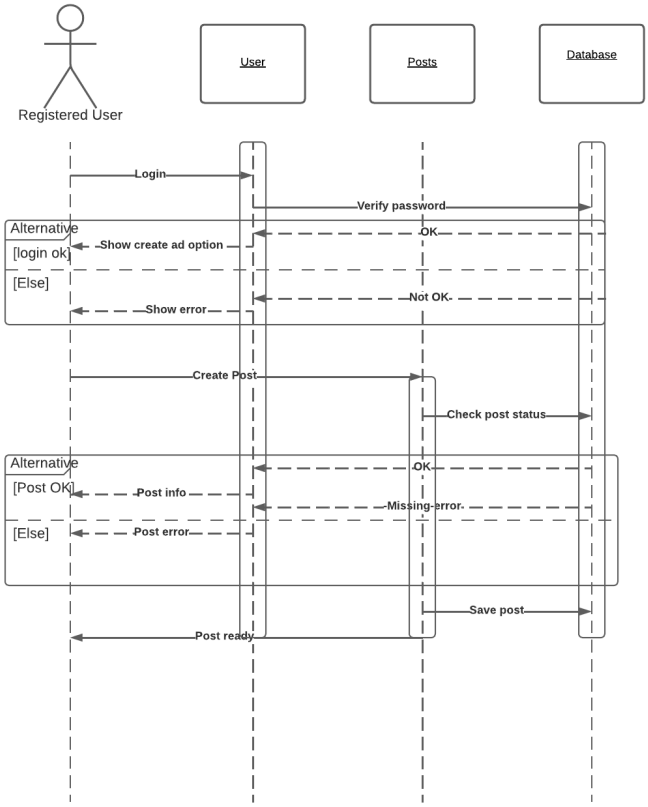

Figure 7: UML sequence diagram representing the process of creating posts.

4.6 View of the implementation

In this subsection we shall have a look at the final implementation of the system. We shall learn more about what the system looks like and all the functionalities it has.

4.6.1 Homepage

The web application is designed in the way that only registered users can use it. Non-registered users can only check the sections about us, contact us and they can click "register" in order to create an account.

A screenshot of the Sport Finder home page is shown in Figure 8.

The home page of the web application is the place where users who want to use the system and all its functionalities will need to log in or create an account.

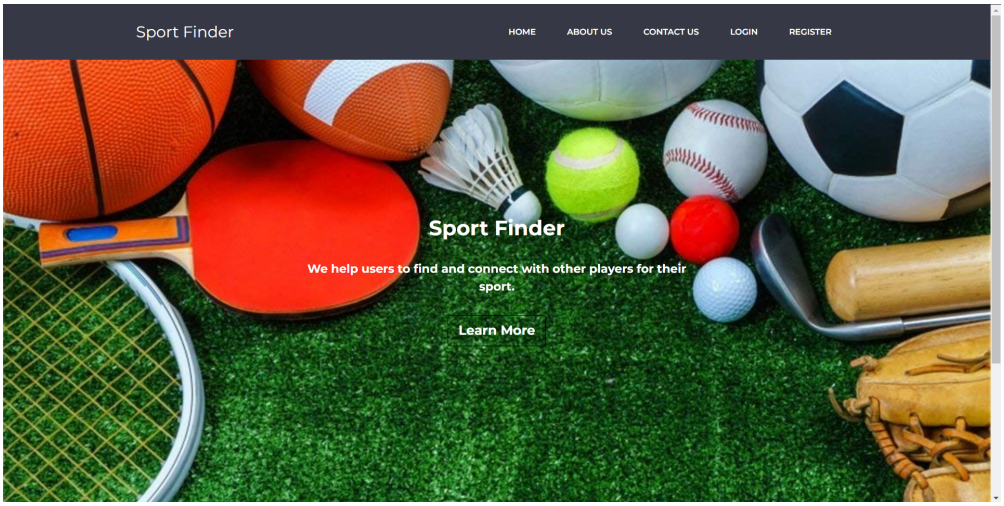

Figure 8: Home page.

4.6.2 Login and Register

Login is done by entering the email address and password. If there are unregistered users, there will be a link that leads to the registration page. A screenshot of the Sport Finder login page is shown in Figure 9.

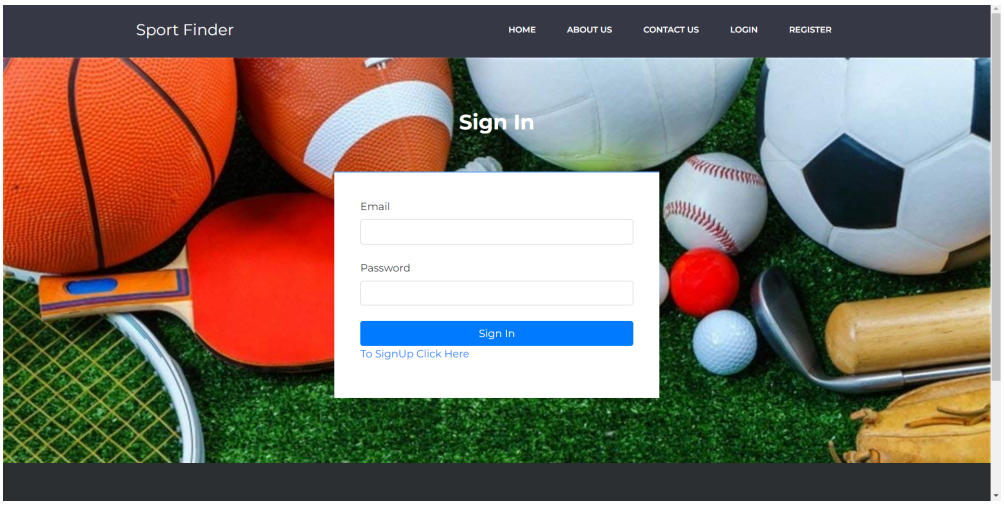

Figure 9: Login page.

For the registration, the user needs to insert their: name, surname, e-mail, and password. After that, they will be added to the database. A screenshot of the Sport Finder registration page is shown in Figure 10.

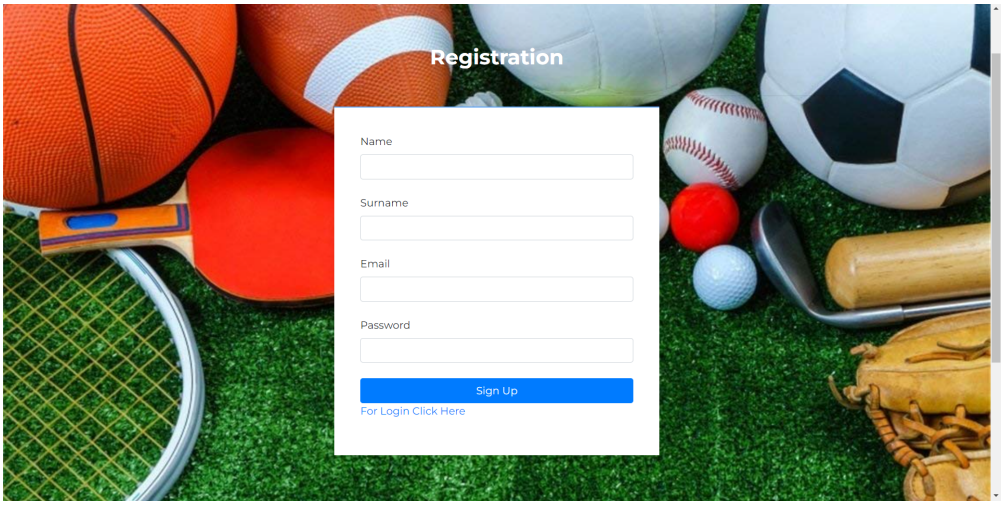

Figure 10: Register page.

4.6.3 Main page for posts

Figure 11 shows the main page of the web application where all posts that are added to the database are displayed. Here, the users can search with a keyword and go through all the posts that match that keyword. If the user wants to connect and play with the owner of the post, there are message windows where they can chat as well as get some more information. We have a rating scale at the bottom of every post where the user can rate. What is more, there is an average rating which can be very useful for other users. A screenshot of the Sport Finder main page for all posts is shown on Figure 11.

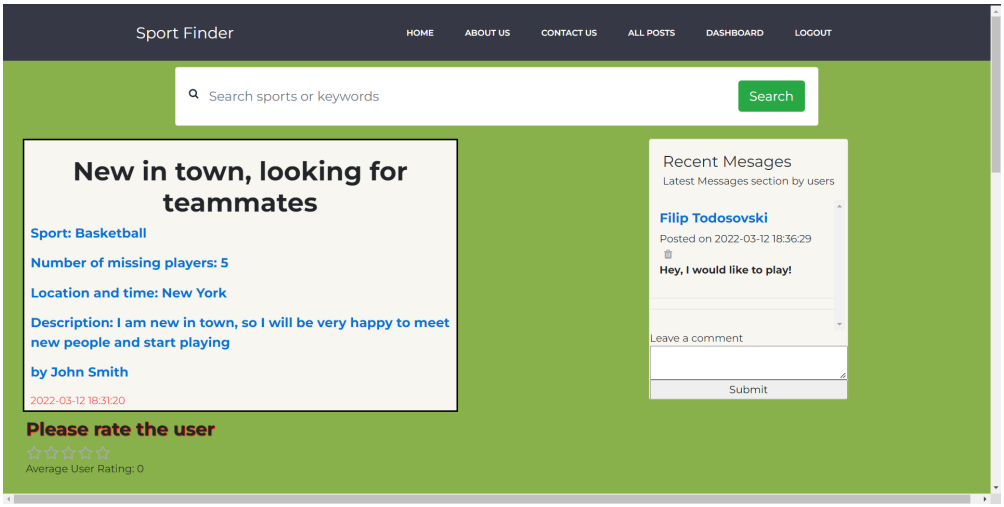

Figure 11: Main page for all posts.

4.6.4 Member area

Figure 12 shows the page where every registered user can create their own posts and have a nice overview of them. Creating a post requires information that will describe what a user needs so that other users will know if that suits them or not. All the created posts are shown on the left side. We also have an "options button" that includes editing or deleting a post. A screenshot of the Sport Finder member area is shown in Figure Figure 12.

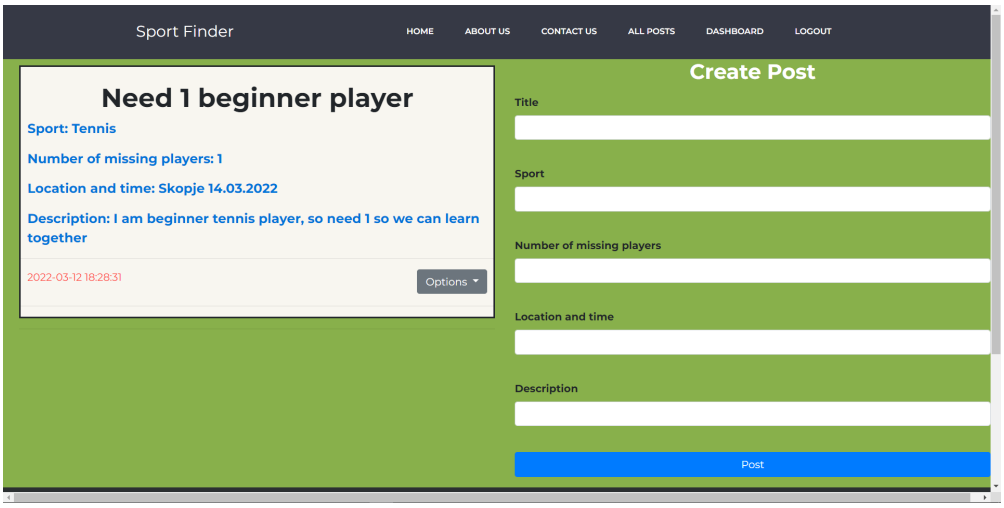

Figure 12: Member area.

4.6.5 Edit

In the edit section, the user has the option to edit a specific part. The user can edit only one single information or multiple information. After clicking "edit", the new information is added to the database. A screenshot of the Sport Finder edit page for posts is shown in Figure 13.

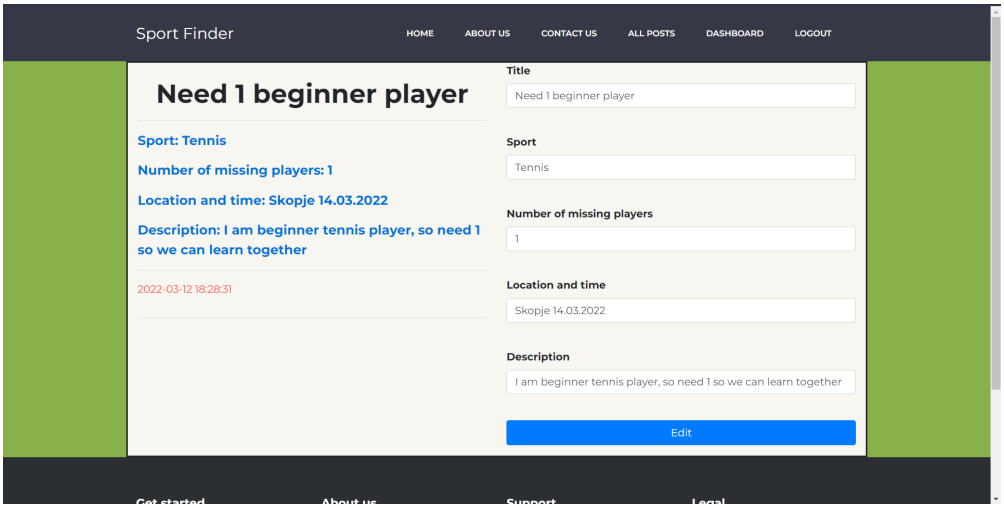

Figure 13: Edit page for posts.

5 Conclusion

In this thesis, we have demonstrated the implementation of a web application that serves as a tool for finding sport teammates that share the same interests. This web application allows users to search through posts that are created by other users in order to find their teammate. Every post contains the information that is required for creating a post. Every user that finds the right post can connect and communicate with other users using the messenger. If the user cannot find the right post, they will have the option to create a new one using the Create post page. In that way, they will be able to specify the required information as well as create their own rules.

This application has been developed primarily due to a scarcity of available applications with such content.

This application can be further developed. One of the possibilities to extend the application is modifying the user profile. This will enable every user to have the option to modify their profile, which will allow better connection among users. Furthermore, when players check somebody's profile and see all the information about their favourite sport, the level they play at, their location, etc. they will be able to decide if they want to keep them in the game or find new players. Eventually, Map API, showing the exact location of each event, will be attached next to the post.

6 Povzetek naloge v slovenskem jeziku

V diplomski nalogi predstavljamo razvoj informacijskega sistema, ki bo pomagal povezati ljudi, ki bi se radi ukvarjali s športom, a nimajo dovolj soigralcev. Najprej predstavimo problem in kako bi ga rešili, kakšna bo naša implementacija in katere tehnologije bomo uporabljali. Ogledamo si tudi nekatere obstoječe sisteme, kako delujejo in katere storitve ponujajo uporabnikom ter kaj bi lahko izboljšali.

V okviru zaključne naloge smo implementirali informacijski sistem, ki bo prikazoval objave igralcev, ki iščejo soigralca. Te objave bodo vsebovale vse potrebne informacije, ki jih morajo vedeti iskalci igre. Najprej smo definirali funkcionalne in nefunkcionalne zahteve, ki opisujejo katere funkcionalnosti bi potreboval naš sistem. Spletno stran si lahko ogleda kdorkoli, uporabljajo pa jo lahko samo registrirani uporabniki. Po registraciji ima uporabnik na voljo vse funkcionalnosti, ki so implementirane v sistemu, kot so ustvarjanje objave, spreminjanje objave, ocenjevanje drugih igralcev, uporaba iskalne vrstice, tako da lahko krmari po objavah in klepeta z drugimi igralci.

Po opredelitvi funkcionalnosti sistema, predstavimo tehnično plat sistema. Katere tehnologije bodo uporabljene in kako se bodo izvajale. Začnemo s predstavitvijo strežniškega dela sistema - kaj to je in kako deluje. Nato na kratko predstavimo programski jezik PHP in ogrodje CodeIgniter, ki pomaga poenostaviti strukturo kode. Poleg tega predstavimo tudi jezika HTML in CSS. Naš sistem mora hraniti različne podatke, ki ijh uporabniki vnesejo. V ta namen smo se odločili za uporabo podatkovne baze MySQL, ki je relacijska baza podatkov in torej podatke shranjuje v tabelah. MySQL je zasnovan za obvladovanje večjega števila delovnih obremenitev. Potek delovanja sistema predstavimo z diagrami UML s katerimi prikažemo, kako bo sistem služil uporabnikom.

Na koncu diplomskega dela predstavimo kako izgleda implementirana spletna stran. Z uporabo zaslonskih posnetkov so prikazane vse funkcionalnosti našega sistema.

7 Bibliography

[1] Carey Wodehouse, *A Beginner's Guide to Back-End Development*, https://www.business2community.com/brandviews/upwork/beginners-guide-back-end -development-01855622. (Viewed on: 15/7/2022.) *(Cited on page 7.)*

[2] Ashwani K, *What is PHP? and How PHP works?*, https://www.devopsschool.com/blog/what-is-php-and-how-php-works/. (Viewed on: 15/7/2022.) *(Cited on page 8.)*

[3] Paul Jackson, *What is CodeIgniter? How Does PHP CI Framework Works?*, https://www.guru99.com/what-is-codeigniter.html. (Viewed on: 15/7/2022.) *(Cited on page 9.)*

[4] Javatpoint, *Why you should use CodeIgniter?*, https://www.javatpoint.com/what-is-codeigniter. (Viewed on: 29/7/2022.) *(Cited on page 9.)*

[5] Carey Wodehouse, *A Beginner's Guide to Front-End Development*, https://www.business2community.com/brandviews/upwork/beginners-guide-front-end-development-01851787. (Viewed on: 15/7/2022.) *(Cited on page 10.)*

[6] *The Importance of HTML*, Cambridge University Press. http://www.cambridgegcsecomputing.org/sites/94/upload/userfiles/216l$_t$eachersnotes$_e$ditedct$_1$50dpi.pdf.($Viewedon$: 15/7/2022.) (*Citedonpage 12*.)

[7] jennifer, *What Programming Languages Can Connect To Mysql Database?*, https://www.rkimball.com/what-programming-languages-can-connect-to-mysql-database/. (Viewed on: 29/7/2022.) *(Cited on page 12.)*

[8] Mark Drake, *What is MySQL?*, https://www.digitalocean.com/community/tutorials/what-is-mysql. (Viewed on: 29/7/2022.) *(Cited on page 12.)*

[9] Richard Peterson, *What is a Database?*, `https://www.guru99.com/introduction-to-database-sql.html`. (Viewed on: 29/7/2022.) *(Cited on page 12.)*