# INTRODUCTION

Mobile payment, also named as mobile money, mobile money transfer and mobile wallet, generally refer to payment services executed on the mobile terminals. Compared to traditional payment with cash, cheque and credits cards, mobile payment show advantages on wide range of applicability, security and convenience. Further, better interactive experience is provided in mobile payment. ~~In terms of anti money laundering and fund fund trackbility, mobile system is more effective, since its stores every transaction record.~~

After filling certain essential personal information and verification process conducted by the system administrator, an user can have an account to login into the system. Inside system, services including export money, import money, transfer, receive and pay are provided. By providing valid merchant information, an user can register to be an merchant, gaining qr code receive function.

Many housing systems in Nigeria today still use the old models to manage their houses, where customers can access the housing information at the house and the information of the customer is written on papers. There is no actual database to save the customer’s details. Customer’s will find houses when he/she is in the city to make reservation/booking.

Developed during 2018 primarily to assist managers, and anyone involved in the management, of student accommodation by setting out the main elements of good management practice. It covered among other matters, health and safety, maintenance and repair, and relationships between managers and student tenants. It was also designed to assist students in understanding both the standards and procedures applying to their accommodation and their obligations as tenants or licensees.

Administration arrangements for the studentsare set out. These are intended to serve the overall purposes of the student helping to maintain and enhance standards and to set out clearly what students may expect of their accommodation and its management. The administration arrangements also provide mechanisms for identifying, and helping remedy, any shortcomings in management of student accommodation.

Different schools will have different numbers of houses: some might have more than 10 houses (with as few as 50 students in each house) or as few as four or fewer (with as many as 200 students in each). In some cases, individual houses can be even larger, as in [McCracken County High School](https://en.wikipedia.org/wiki/McCracken_County_High_School" \o "McCracken County High School) in the U.S. state of Kentucky, whose five houses have nearly 400 students each.Facilities, such as [pastoral care](https://en.wikipedia.org/wiki/Pastoral_care" \o "Pastoral care), may be provided on a house basis to a greater or lesser extent depending on the type of school. Historically, the house system was associated with established [public schools](https://en.wikipedia.org/wiki/Public_school_(UK)" \o "Public school (UK)) in England, especially full [boarding schools](https://en.wikipedia.org/wiki/Boarding_school" \o "Boarding school), where a 'house' referred to a [boarding house](https://en.wikipedia.org/wiki/Boarding_house" \o "Boarding house) at the school. In modern times, in both day and boarding schools, the word 'house' may refer only to a grouping of pupils, rather than to a particular building.

Houses may be named after [saints](https://en.wikipedia.org/wiki/Saint" \o "Saint), famous historical alumni or notable regional topics (e.g. in international schools, houses are sometimes named in honor of local celebrities). Other more arbitrary names animal names or colours, for example are also often used. Houses are also often referred to by the original name of the building or by the name or initials of the teacher in charge of the house ([housemistress](https://en.wikipedia.org/wiki/Housemistress" \o "Housemistress) or [housemaster](https://en.wikipedia.org/wiki/Housemaster" \o "Housemaster)). Each house will usually also be identified by its own symbol, logo, or colours.

At co-educational boarding schools, there may be separate houses for boys and girls, as at the [Lawrenceville School](https://en.wikipedia.org/wiki/Lawrenceville_School" \o "Lawrenceville School), whose house system is itself based on that of [Rugby School](https://en.wikipedia.org/wiki/Rugby_School" \o "Rugby School). Students may also be grouped by year groups or status as boarders or day students. At [Winchester College](https://en.wikipedia.org/wiki/Winchester_College" \o "Winchester College) and [Eton College](https://en.wikipedia.org/wiki/Eton_College" \o "Eton College), there is a separate house for foundation scholars. Where the school has boarders and day pupils like the [King's School, Canterbury](https://en.wikipedia.org/wiki/King%27s_School,_Canterbury" \o "King's School, Canterbury), they will often be allocated to separate houses. There have also been cases, for example at [Cheltenham College](https://en.wikipedia.org/wiki/Cheltenham_College" \o "Cheltenham College), of pupils being allocated to different houses according to their religion. At traditional full boarding schools such as [Radley College](https://en.wikipedia.org/wiki/Radley_College" \o "Radley College) and [Harrow School](https://en.wikipedia.org/wiki/Harrow_School" \o "Harrow School), students are grouped by boarding house.

The goal of this project web development is to create a new application called student housing search system. This allows students to receive information about the availability of apartments or rooms. This web application allows users to easily access all the information related to apartment rentals.

# ANALYSIS OF SUBJECT AREA

## 1.1 Aim And Objectives Of The Project

The purpose of the student housing search system is to speed up the search for housing, to be able to handle all operations according to the desired criteria Such as the choice of the region, the availability of the subway, the number of rooms, the cost and so on. This automated system will significantly reduce the time spent searching for student accommodation. It saves all important information stored safely in a database that can be restored from time to time. The system is convenient and easy to use. Efficiency is improved for everyone involved in automated systems. Saving data is easier.

The goal of this project web development is to create a new application called student housing search system. This allows students to receive information about the availability of apartments or rooms. This web application allows users to easily access all the information related to apartment rentals.

## 1.2 Organizational Structure and Design of Housing System

An organizational structure is a pattern or network of relationships that exist among positions. It is a pattern of relationships that has been generated through a conscious planning process. Key executives typically decide upon the basic pattern of structure that in their opinion will be most appropriate for themselves, their work, and organization goals.

1.2.1 The Importance of Organizational Structure

* Clear definition of authority, responsibility relationship facilities better understanding of the objectives and the policies of the enterprise.
* Organizational structure lays down both channels and the patterns of communication. It facilitates proper administration.
* It helps to coordinate activities of the component parts in order to facilitate the realization of the goals of the organization.
* It helps in growth and diversification of the activities of an organization.
* Workers, participation in organization increases their cooperation and improves their will to work. It stimulates initiation and creative thinking.
* Implementation of policies and the achievement of the goals become easier.
* It prevents duplication of functions and makes it possible to achieve maximum production with minimum efforts.

1.2.2Approaches to Organization, Structure and Management

Organization structure, the process of management and the behavior of people at work are inextricably linked. Underlining the development of management theory and practice are contrasting ideas on structural design and attitudes towards people. Identification of major trends in the study of organizations and management will help to provide a perspective on concepts and ideas.

1.2.3 The Value of Organization Charts

It is usual for the structure of the housing to be depicted in the form of an organization chart. This can be very useful in providing a pictorial presentation of the structural framework of the housing and its main area of activities. It is helpful, for example, as part of a staff induction manual. The chart may also be used as a basis for the analysis and review of structure, for training and management succession, and for formulating changes.

An organization chart may show, at a given point in time, how work is divided, spans of control, the levels of authority, lines of communication, and formal relationships. But charts vary greatly. Some are intended to give a minimal amount of information, perhaps for example only an outline of the management structure of the housing.

Every housing, whether it’s big or small, needs an organizational structure to carry out its daily operations. It is used to help divide tasks, specify the job for each department, and delegate authority within and among departments. Effective job specifications will increase work productivity and efficiency. Each housing organizes workforce in different ways. Here shows a medium size housing organizational structure. See figure 1.1 below.

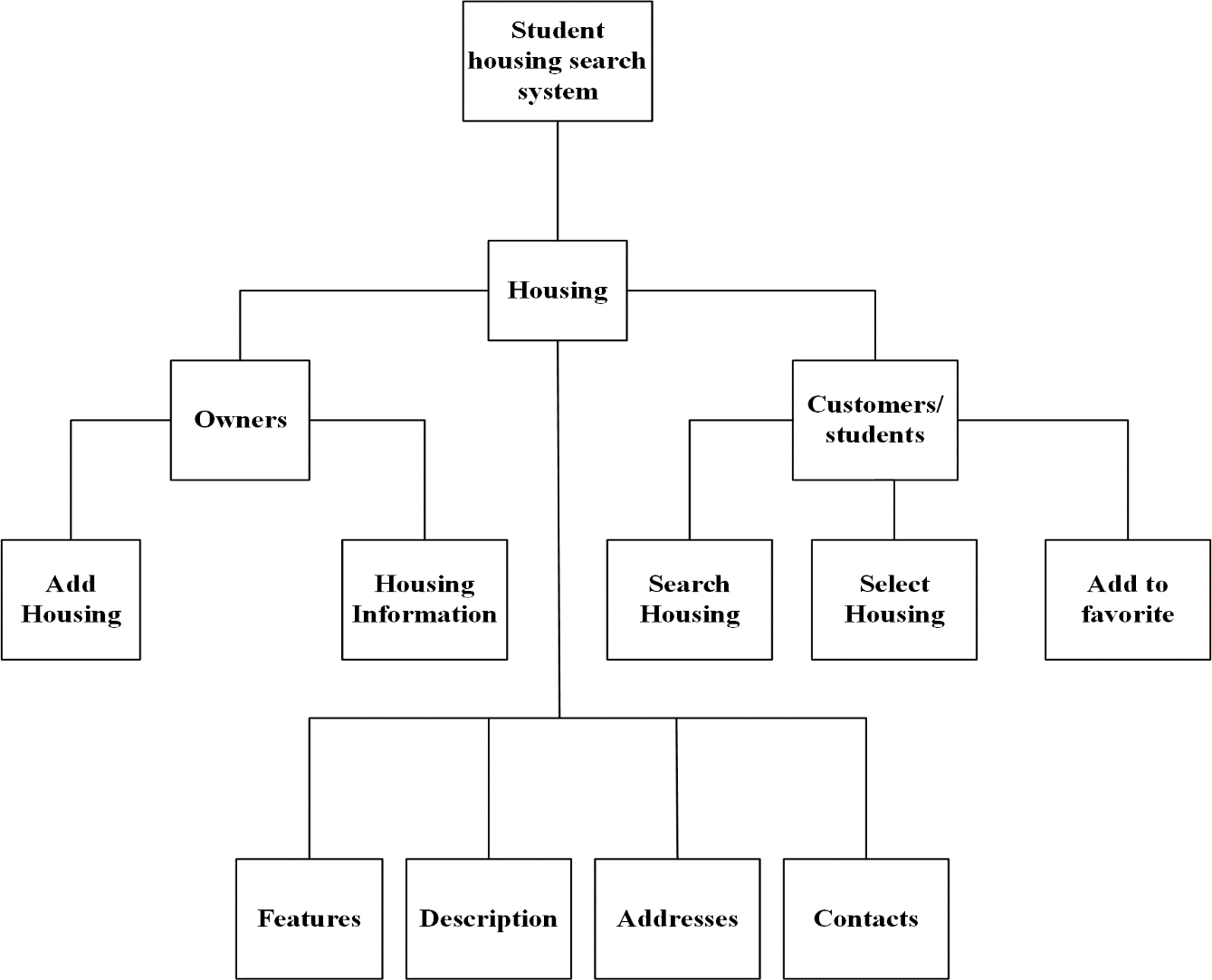


Figure 1.1 – Organizational Structure Student Housing Search System

The structure is a system from which housing owners can add information concerning their houses or students in search of housing can input specific search criteria.

Existing housing owners can update their houses and contact information, add new property if any and New owners can put their housing for lease ,rent etc.

Customers/students in search of housing input their specific search criteria: area, region, price etc.

## 1.3 Scope of the study

Higher (and further) education establishments have distinctive management arrangements including independent governing bodies. They are subject to both internal and external independent audit. The relationship between university or college authorities and students as tenants, or licensees, therefore differs in some important respects from the landlord/tenant relationship in the privately rented sector. Moreover, a range of statutory requirements applies to provided student housing, some general but some specific.

The basic requirements are those that would need to be met were the relevant properties to be subject to licensing under the Act. They are concerned with fire safety, maintenance of gas and electricity installations, water supply and drainage, toilets, sinks, washbasins, installations for storing food, disposal of refuse and litter, maintenance of common areas (e.g. yards, gardens), windows and ventilation, repair of internal structure and post boxes. Additional basic requirements are that tenants are to be advised on action to be taken in case of emergencies and that managers must be allowed reasonable access to the premises.

## 1.4 Existing Student Housing Information System

Student Housing involves various operations of the house like Booking or reservation of rooms, cancellation of the rooms, cash billing etc.

The existing system is a manual one and there are lots of issues like erroneous data, slow process, lack of security etc. Finding out the final payment amount completely relies on the house/hostel manager and if he is absent, it takes a long time to find out the details and is prone to errors.

1.4.1Drawbacks of the Existing System

In our existing system all the booking of rooms in student housingare done manually, so it was more time taking for an executive and customer to book, search rooms and collect payment in the house.

Another major disadvantage is that to preparing the list of all the rooms available in the houses will take more time, currently it is doing as a one day process for verifying all records. After conducting the feasibility study we decided to make the manual Student housing search system to be computerized.

We had analyzed different student housing search system and after that we had find the given problems in their system that are mentioned below:

* manual record keeping is not consistent as some errors can creep in while writing manually;
* it is difficult to keep record of room availability and large number of customers.
* more labor and time is required in this current system;
* managing room availability in order is bit difficult, so it’s not possible for executive to search the room;
* inaccuracy of records;
* no central database can be created as information is not available in database;
* more funds and paper is wasted to keep the manual record of available rooms and customers.

Enhanced system i.e. Student HousingSearch System in this system developer had tried to remove all the problems that has been faced while using manual housing booking system. In manual system customers find difficult to find the houses that are available but in current system they can easily find the details on a single click in this system. Calculating payment is easy in this system and most important thing is that a single person (Admin/User) can handle whole system without facing any problems through this system labor cost has been reduced. This system is time efficient, and full of accuracy, through this system admin has not to think about any error that was facing during the manual system. Below are some examples of the existing systems.

1.4.2StarRez Housing System

StarRez offers the most complete, easy-to-use, and fully integrated solution for student housing and accommodation providers both small and large. StarRez is a comprehensive student housing solution including: online housing applications, online roommate and room self-selection, roommate matching, core staff administrative functions such as bulk room assignments/auto-allocation, billing, reporting, email, and mail merge, as well as, complete integration with other campus systems.

1.4.3Ohio Stater Housing System

The Ohio Stater offers students alternative housing options to campus dorms. With the same convenience of a close proximity to classes that dorm living provides, Ohio Stater Student Housing also has the privacy and security benefits of an apartment

It is affordable and the rooms are fully-furnished with central heating and air conditioning and all of your utilities are included with your rent, including internet and basic cable. You can even prepare your own meals in one of the common kitchens to save money on eating out.

## 1.5 Proposed Student Housing Search System

During the past several decades’ personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources:

* a computer based management system is designed to handle all the primary information required to calculate monthly statements. Separate database is maintained to handle all the details required for the correct statement calculation and generation;
* this project intends to introduce more user friendliness in the various activities such as record updation, maintenance, and searching;
* the searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification of that customer;
* similarly, record maintenance and updation can also be accomplished by using the identification of the customer with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date;
* the entire information has maintained in the database or Files and whoever wants to retrieve can’t retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file;
* the main objective of the entire activity is to automate the process of day to dayactivities of Houses like:
* Admission of a New Student;
* Assign a room according to student’s demand;
* Checkout of a computer and releasing the room;
* Finally compute the bill etc;
* Packages available;
* Advance online bookings;
* Online Cancellation;
* List of Regular customers/students;
* Email facility;
* Feedbacks.

The proposed system is to facilitate easy management and administration of the houses with capabilities to do Booking or reservations of the rooms, Cancellation of the rooms, Cash billing, Room service, Restaurant service, Restaurant billing, Total billing, Travels arrangement etc. using the student housing search system.

This system is designed to help students find houses during their period of study. As the internet grow bigger every day, this system will make students, tourist and other individuals trying to travel make booking online rather than going to the country and then starting to move from one hotel/student houseto the other in search of private houses.

Online student housing search system website allows you to browse through endless possibilities, and even offers houses that are available in different cities of your choice. What is even more useful is the ability to compare houses, similar or not online. You can search through multiple housing at the same time, comparing material quality, sizes and pricing simultaneously.

Say 'goodbye' to the days when you drove endlessly looking for housing to check in and relax. Online housing transactions occur instantly saving you time to get your booking done.

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of booking, reservation and payment on the web is becoming a common place.

The objective of this project is to develop a multipurpose online student housing search system where several houses can be placed online to easy access to students who will make reservation, booking get info on any student housing system from the comfort of home through the Internet.

1.5.1Advantages of housing system

The following are the objectives and highlights of the proposed system.Less Expensive: Have you ever advertised your business through various forms such as printed media, radio, and television or by other means? It’s expensive! Investing in advertising is necessary, but it takes a lot of money. Having a website will make promoting your company less expensive. Many versions of offline advertising available on the Internet are sometimes free.

**Advertising:** A website is more environmental friendly when it comes to advertising and marketing. There are lots of ways to advertise your products or services through the Internet. One example is Facebook ads, an advertising feature offered through Facebook. Another one is called SEO. This is a major advantage for your business. Having a good SEO service provider can boost the ranking of your website which quickly results in increased sales and higher profits.

**Satisfaction:** Having a website will be more convenient for your customers and leads. Make it easy for your customers to purchase from you! Many will be more likely to visit your website, rather than driving a car to your physical location and browsing for your products. From a customer’s point of view, it’s better for them if they don’t have to ask anything. They can just find what they’re looking for on your online site.

**Increase Customers:** Most businesses have local popularity, but what about potential customers outside their city? A website can help you generate more customers. Not just outside your city, but also worldwide. The internet offers a global community. With a website, your business will be visible around the world.

**Accessibility:** Have you ever experienced having to turn customers away because it’s closing time? Well, you don’t have to close the doors of your website. An online site can be visited any time of the day or night. People will look to your site instead of going to your shop because it is more accessible. Just make sure to post enough information about your products and services.

**Access to Info:** Did you know that if you own a website, you can actually track everything that is happening on it? You can even look for information that will tell you how many people visited your site, or how many people messaged or emailed you. You can access the progress of your website and view all its pages. You can even make an update anytime, making it much less expensive than printed material.

**Better Relationship:** Having a website can build better relationships with your customers. You can send messages instantly to your customers through email. Also, your customers can review your products online and can also leave feedback for you and your business. It’s best to always send your customer a message. This is essential for building a good relationship with them. You can even give them more information about your business through messages or emails.

**Increase Sales:** If you are a business owner, more visitors lead to more potential sales. That’s how your website will help you. You can drive more people to your site by consistently updating and promoting the contents of your site. The more informative your site is, the greater the possibility of increasing your sales.

**Opportunity:** A website gives you the opportunity to prove your credibility. You have to tell your customers why you deserve their trust through your website. This can earn positive feedback for your service and products. Also, your website serves as a place for a potential investor to explore what your business is about and what it can do in the future.

**Long Term Clients:** What do you think is the difference between client and a customer? Well, a customer is the one who walks in and buys something and that’s it. A client is your regular customer. He is buying your products or services daily or contractually. Having a website gives you a chance to gain more clients that can help your business grow.

# 2 DESIGNED APPLICATION STUCTURE OF THE student housing search system

## 2.1 Selection of Design Methodology

As per the project management there should be a proper selection of the methodology so on that contrast we have to select the methodology for the project i.e. Student Housing Search System. The method being used in developing the system is the System Development Life Cycle (SDLC). The SDLC process includes project identification and selection, project introduction and planning, analysis, design, implementation and maintenance.

* + 1. Purpose of Methodology
* it provides us with a framework for planning the project;
* the quality of a software development effort is improved using selected process techniques.

**2.1.2** Structure of the methodology

The general structure of life cycle model is shown in the figure 2.1.

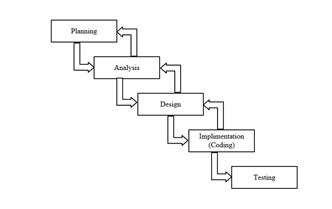


Figure 2.1 – Methodological Structure

Each phase produces deliverables required by the next phase in the life cycle. Some phases are also inter-related with other phase. Planning is done in the first and analysis of the system is continued after that. Analysis of the system is being translated into design. Code is produced during implementation that is driven by the design. Testing verifies the deliverable of the implementation phase.

**Planning**: It is the very first phase of the system, in which we decide what the task is for and what all things we have to do for completion of the project in an easier and healthy way.

**Analysis**: It is second and also considered as an important phase of the life cycle model. In this phase, the existing system is studied by collecting the information through the Internet and analyzed the information to get alternatives for the use of proposed system.

**Design**: This is where the details on how the system will work are produced. Architecture, including hardware and software, communication are all part of the deliverables of a design phase.

**Implementation**: This is the longest phase of the software development life cycle as the code is produced from the deliverables of the design phase during implementation.

**Testing**: During testing of the system, the implementation code is tested to make sure that the product is actually solving the needs addressed and gathering the other information. Unit testing and system testing are done during this phase.

## Implementation of the Student Housing Search System

Implementing a machine learning algorithm in code can teach you a lot about the algorithm and how it works.In this post you will learn how to be effective at implementing machine learning algorithms and how to maximize your learning from these projects.

You can use the implementation of machine learning algorithms as a strategy for learning about applied machine learning. You can also carve out a niche and skills in algorithm implementation.

Implementing a machine learning algorithm will give you a deep and practical appreciation for how the algorithm works. This knowledge can also help you to internalize the mathematical description of the algorithm by thinking of the vectors and matrices as arrays and the computational intuitions for the transformations on those structures.

There are numerous micro-decisions required when implementing a machine learning algorithm and these decisions are often missing from the formal algorithm descriptions. Learning and parameterizing these decisions can quickly catapult you to intermediate and advanced level of understanding of a given method, as relatively few people make the time to implement some of the more complex algorithms as a learning exercise.

The algorithm has the following properties:

* discreteness. this property is that the algorithm must represent the process of solving a problem as a sequential execution simple steps. to perform each step of the algorithm, the final time interval, that is, the transformation of the original data into the result is discrete in time;
* definition. each algorithm rule must be clear, single-valued;
* efficiency. the algorithm should lead to a solution for finite number of steps;
* the mass. the algorithm for solving the problem is developed in general form, that is, it must be applicable to a certain class of problems, differing only in the initial data;
* correctness. the algorithm is correct if its execution gives correct results of solving the task.

There are different means to make algorithms. The choice of the tool is determined by the type of the algorithm to be executed. The following are ways of recording algorithms:

* Verbal: when the algorithm is described in human language.
* Symbolic, when the algorithm is described by a set of symbols.
* Graphical, when the algorithm is described using a set of graphic images.Common methods of recording are graphical recording using flowcharts.
* Versatility. The algorithm can be used for an entire class tasks, to different sets of initial data.

In figure 2.2 below is a flowchart of customer making reservation in the system.

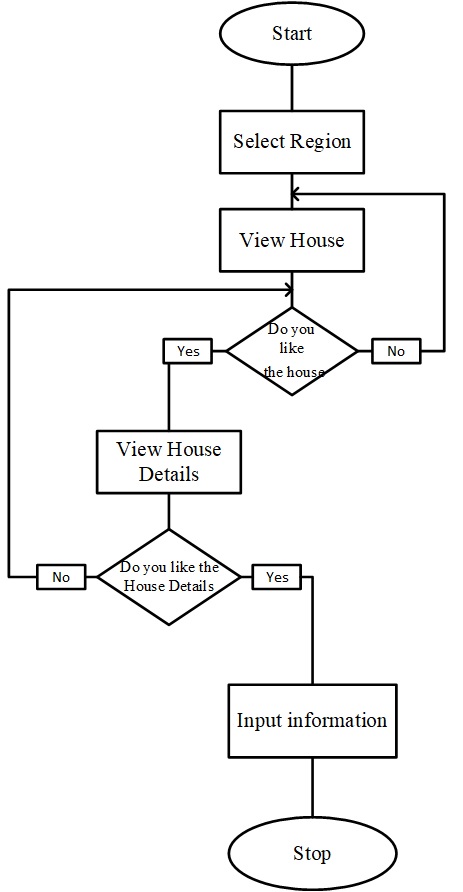


Figure 2.2 – System flowchart: student housing searchsystem

The description of the algorithm using flowcharts is done by drawing a sequence of geometric shapes, each of which implies the execution of a certain action algorithm.

Block diagram is a graphic representation of the logical structure of the algorithm, in which each stage of the information processing process is represented in the form of geometric figures, the configuration of which determines the nature of the indicated actions. Conditional graphics used in the construction of schemes are called symbols.

**2.3****IDEF0 DIAGRAM**

IDEF0 (Integration Definition for Function Modeling) is a function modeling methodology for describing manufacturing functions, which offers a functional modeling language for the analysis, development, reengineering, and integration of information systems; business processes; or software engineering analysis.IDEF0 is part of the IDEF family of modeling languages in the field of software engineering, and is built on the functional modeling language Structured Analysis and Design Technique (SADT).

The IDEF0 Functional Modeling method is designed to model the decisions, actions, and activities of an organization or system. In its original form, IDEF0 includes both a definition of a graphical modeling language (syntax and semantics) and a description of a comprehensive methodology for developing models. IDEF0 should assist in organizing system analysis and promote effective communication between the analyst and the student through simplified graphical devices.The diagram below shows the student housing search system in figure 2.3 below.

The four arrow types, Inputs, Controls, Outputs and Mechanisms are collectively called ICOM and IDEF is an abbreviation of ICOM DEFinition. There is a zero on IDEF0 because there are a number of additional IDEF standards.The side of the activity box, which they touch, identifies the different ICOM arrows. Thus inputs are on the left, controls at the top, outputs on the right and mechanisms at the bottom.

****

Figure 2.3 – Context diagram for student housing search system

* This can make diagrams a bit more difficult to draw but can make them more easily readable.
* The system inputs are; Search information, select information, select information.
* The system outputs are: Apartment not found and apartment found.
* The system control is the searching instruction.

## 2.3UML Use Case and Sequence Diagrams of the Student Housing Search System

For the description of the algorithm used in the system diagrams,UML is used.The UML (Unified Modeling Language) is a standard visual modeling language intended to be used for modeling business and similar processes, analysis, design, and implementation of software-based systems. UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.



Figure 2.4 – Decomposition diagram of the System

2.3.1 Use Case diagram

**Use case diagrams**are usually referred to asbehavior diagramsused to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external user of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. [3]

In this project, there are two actors, which are the Administrator and the customers (figure 2.5).

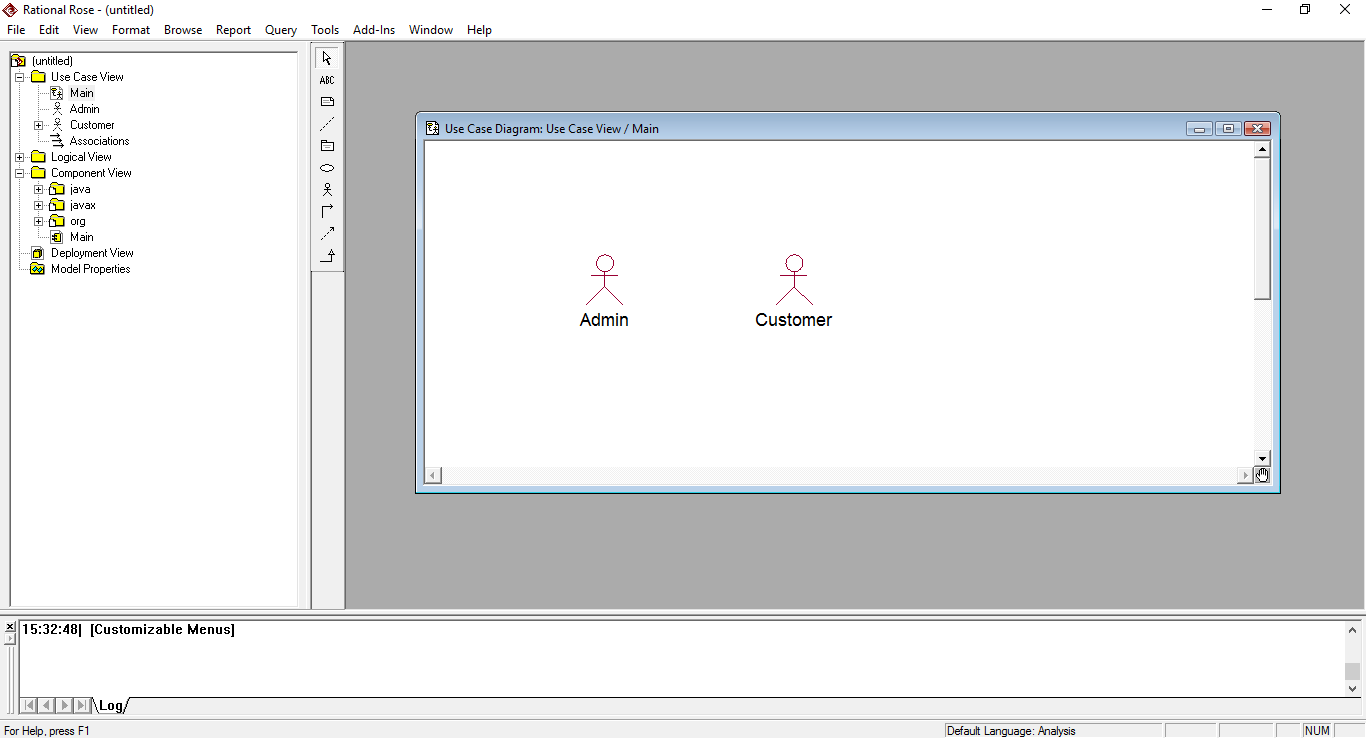


Figure 2.5 – Actors of the System

The customer can view the content of the website: the list of States and Cities, list of Houses in different cities, contact the houses by sending messages and view housing information.

In this system the customer does not need to login to book any house. The customer can make comments in the website, check for housing availability, make payment, Search city, Search State, and cancel booking (figure 2.4).

****

Figure 2.6– Use Case Diagram

2.3.2 Sequence diagram

Sequence diagrams show how objects communicate with each other in terms of a temporal sequence of messages. The time flow is the most visible aspect in these diagrams, as messages are sequenced according to a vertical timeline and also the lifespan of objects associated to those messages is reported.

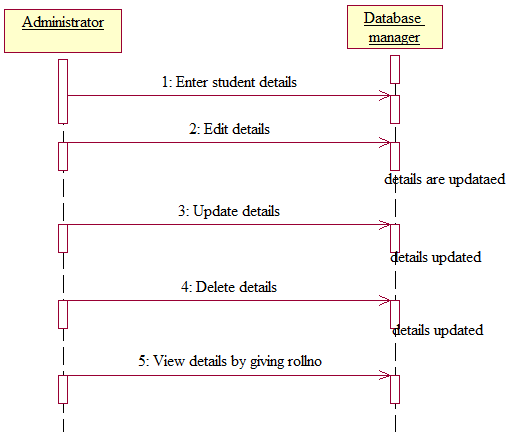


Figure 2.7– Sequence diagram for Admin and Database.

The diagram above shows the interaction between Administrator and Database manager.

The admin requesting information about students:

* enter student details – review student details;
* edit details – edit the student details branch, address, roll no etc;
* update details – update student details;
* delete details – delete details of student;
* view details by giving roll no.

2.3.3 Collaboration diagram

A collaboration diagram shows the interactions between objects or classes in terms of links and messages that flow through the links. This describes at the same time some kind of static structure (links and nodes) and dynamic behavior (messages) of the system.



Figure 2.8– Collaboration for Admin and Database.

2.3.4Activity diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

The diagram below shows the processes of using the student housing system. When loading the page, a user or admin enter his/ her credentials using the login column, then searches for a house using the view house option. The admin is allowed to upload, delete, and edit new houses. Using the user account, he can do likewise, but isn’t permitted to delete, edit an already existing uploaded houses put up by the admin

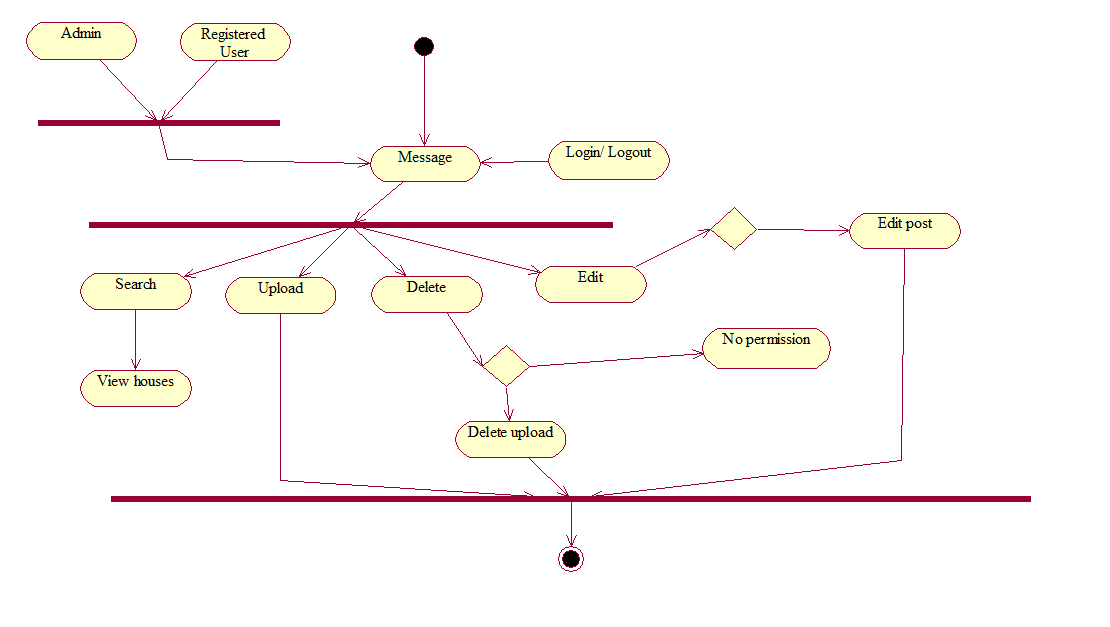


Figure 2.9– Activity diagram for Student housing system processes.

## 2.4 Design and Description of the Database System

A database is a more complex object; it is a collection of interrelated stored data that serves the needs of multiple users within one or more organizations, that is, interrelated collections of many different types of tables. The motivations for using databases rather than files include greater availability to a diverse set of users, integration of data for easier access to and updating of complex transactions, and less redundancy of data.

Database design - is process of creating a database schema, and determining the necessary integrity constraints.

The main objectives of the database design are:

* to secure the database with all the necessary information;
* ensuring the possibility of obtaining all the necessary data requests;
* reducing redundancy and duplication of data;
* ensuring data integrity (correctness of their content): elimination of contradictions in the content of the data, with the exception of their loss. The database reflects information about a specific subject region.

A database schema represents the logical configuration of all or part of a relational database. It can exist both as a visual representation and as a set of formulas known as integrity constraints that govern a database. These formulas are expressed in a data definition language, such as SQL. As part of a data dictionary, a database schema indicates how the entities that make up the database relate to one another, including tables, views, stored procedures, and more (figure 2.8).

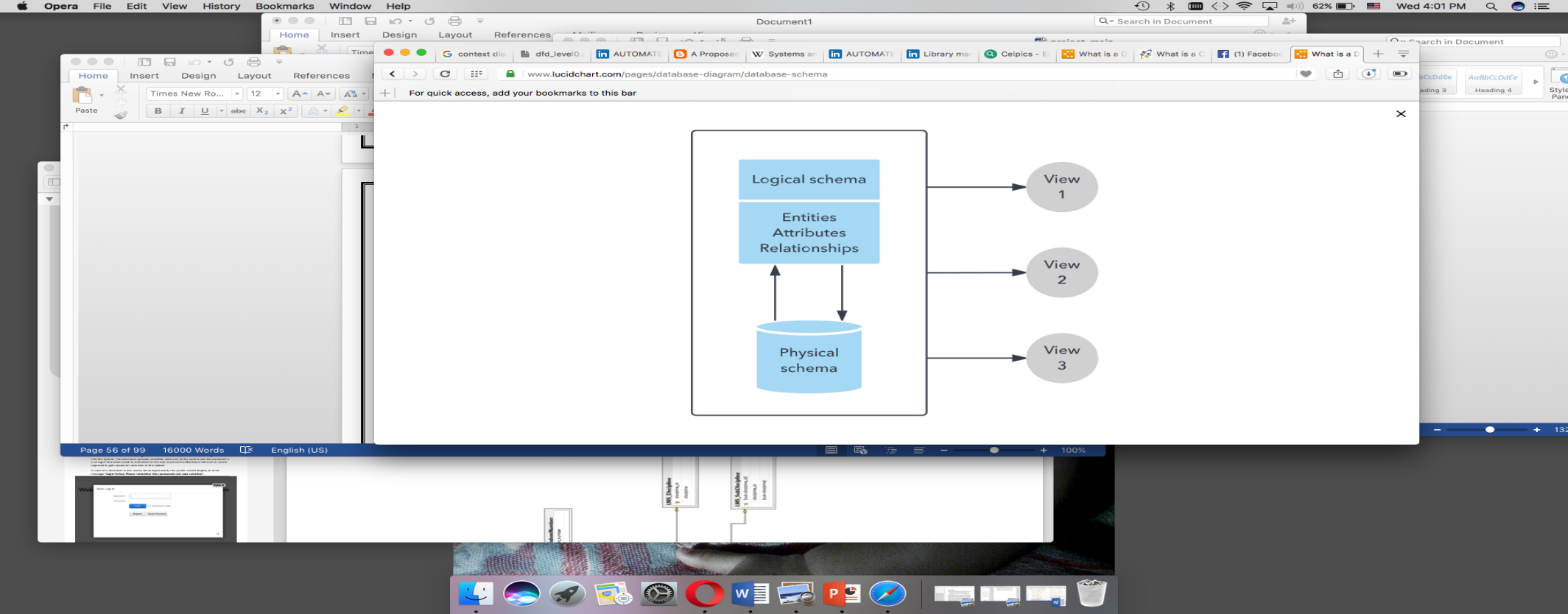


Figure 2.10 – Database Structure

Typically, a database designer creates a database schema to help programmers whose software will interact with the database. The process of creating a database schema is called data modelling. When following the three-schema approach to database design, this step would follow the creation of a conceptual schema. Conceptual schemas focus on an organization’s informational needs rather than the structure of a database.

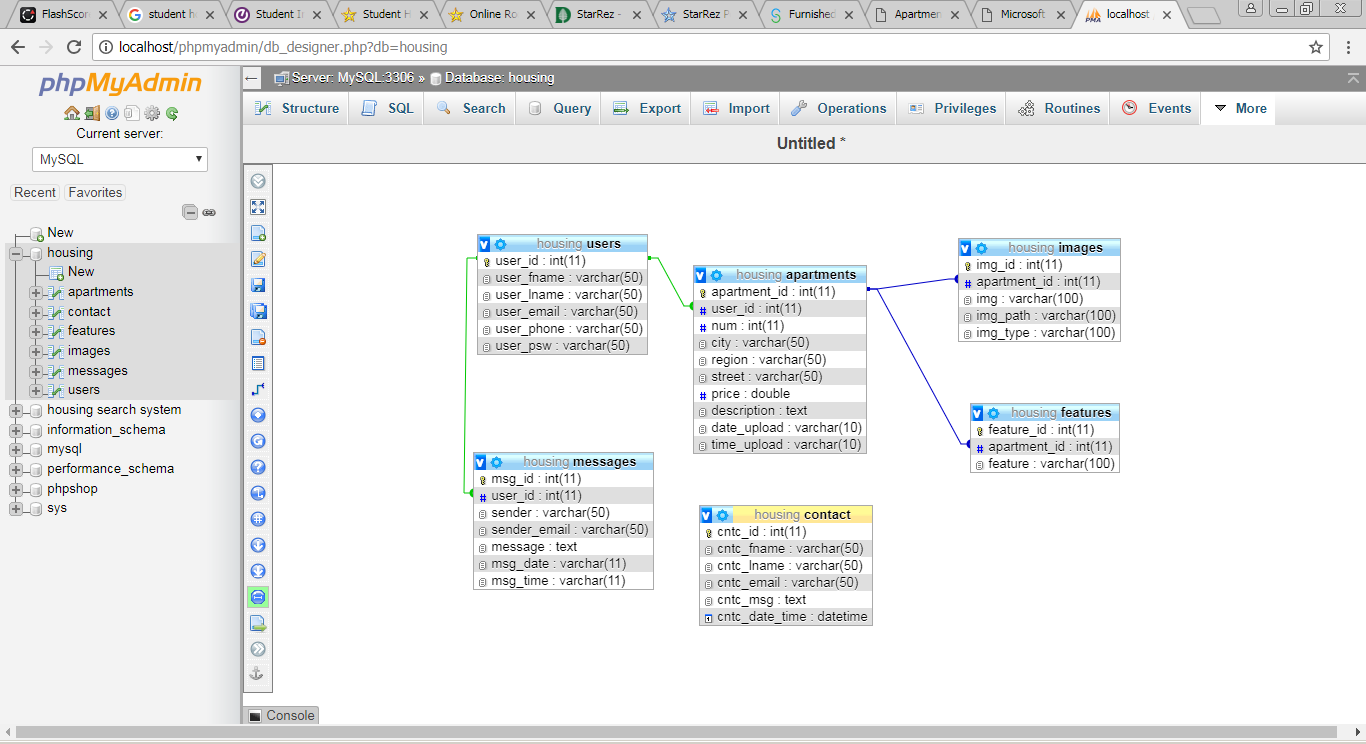


Figure 2.11 – Database Structure

The developed database system consist of 6 tables, The table User contains user\_id from where the user will enter username and password. With User Id as the primary key.The structure of the user is shown in table 2.1.

Table 2.1 – The structure of the tableUser.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Length | Allow Nulls | Name Description |
| User\_id | Int | 20 | Not Null | User identity |
| User\_fname | Varchar | 20 | Not Null | The first name of the user |
| User\_lname | Varchar | 20 | Not Null | The last name of the user |
| User\_email | Varchar | 30 | Not Null | Email address of the user |
| User\_phone | Varchar | 15 | Not Null | Phone number of the user |
| User\_pws | Varchar | 10 | Not Null | Password of the user |

The table Message contains messages sent. With message\_id as the primary key and user\_id as the foreign key from. The structure of the message is shown in table 2.2.

Table 2.2 – The structure of the table Message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Length | Allow Nulls | Name Description |
| MessageId | Int | 20 | Not Null | Message identity |
| UserId | Int | 20 | Not Null | User identity |
| Sender | Varchar | 20 | Not Null | The person sending the message |
| SenderEmail | Varchar | 30 | Not Null | Email address of the sender |
| Message | Text |  | Not Null | Message sent |
| MsgDate | Varchar | 11 | Not Null | Message date |
| MsgTime | Varchar | 11 | Not Null | Message time |

The table Image contains information of the housing images. The Primary Key in this table is the Image Id. The structure of the tableImage is shown in table 2.3.

Table 2.3 – The structure of the table Image.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Length | Allow Nulls | Name Description |
| ImageId | Int | 10 | Not Null | Image Identity |
| ApartmentId | Varchar | 20 | Not Null | Apartment Identity |
| Img | Int | 10 | Not Null | Image |
| ImagePath | Varchar | 100 | Not Null | Location of the Image |
| Img Type | Varchar | 100 | Not Null | Image Type |

The table Apartment contains information of the different apartments. The Primary Key in this table is the Apartment Id. The structure of the table Image is shown in table 2.4.

Table 2.4 – The structure of the tableApartment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Length | Allow Nulls | Name Description |
| Apartment Id | Int | 11 | Not Null | Apartment Identity |
| User Id | Int | 11 | Not Null | User Identity |
| Number | Int | 11 | Not Null | Number of Apartment |
| City | Varchar | 50 | Not Null | City of the Apartment |
| Region | Varchar | 50 | Not Null | Region of the Apartment |
| Street | Varchar | 50 | Not Null | Street of the Apartment |
| Price | Double |  | Not Null | Price of the Apartment |
| Description | Text |  | Not Null | Description of the Apartment |
| Date Upload | Varchar | 10 | Not Null | Date upload of the Apartment |
| Time Upload | Varchar | 10 | Not Null | Time upload of the Apartment |

The table Feature contains information of the different features in the apartments. The Primary Key in this table is the Feature Id. The structure of the table Image is shown in table 2.6.

Table 2.6 – The structure of the table Feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Length | Allow Nulls | Name Description |
| Feature Id | Int | 20 | Not Null | Admin username |
| ApartmentId | Int | 20 | Not Null | Admin password |

The table Contact contains the information about customer that contacted the housing owner, with the ContactId as the Primary Key.The structure of the tableContact is shown in Table 2.5.

Table 2.5 – The structure of the table Contact.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **DataType** | **Length** | **AllowNulls** | **Name Description** |
| Cnct Id | Int | 11 | Not Null | Contact Identity |
| Cnct Fname | Varchar | 50 | Not Null | Contact First Name |
| Cnct Lname | Varchar | 50 | Not Null | Contact Last Name |
| Cnct Email | Varchar | 50 | Not Null | Contact Email |
| Cnct Msg | Text |  | Not Null | Contact Message |
| Cnct Dt Tm | DateTime |  | Not null | Contact Date and Time |

## 2.5Hardware and System SoftwareRequirements

Hardware and System Software requirements gives us the insight for developing the system and the minimum requirements needed to run the developed system.

**Software Requirement**:

* Operating System: Windows XP/Vista/2000/Windows 8/10, Linux;
* Presentation layer: CSS, HTML, JQuery’s, PHP, Javascript ;
* Database: MySql;

**Hardware Requirements:**

The minimum requirements;

* Processor: Standard processor with a speed of 1.6 GHz;
* RAM:2 GB RAM or more;
* Hard Disk: 20 GB or more;
* Monitor:Standard color monitor;
* Keyboard: Standard keyboard;
* Mouse:Standard mouse;

## 2.6Ergonomics

The interaction of users with the systemwas carried out using PHP and HTML. The interface of the system should be understandable and convenient, the system should not be overloaded with graphics and should provide a quick display of the screen. Navigation elements must be made in a user-friendly form. The means for updating information must satisfy the accepted agreements in terms of the use of function keys, operating modes, search, and use of the window system. The interface should correspond to modern Ergonomic requirements and provide easy access to the main functions and operations of the system.

The system should ensure correct handling of emergencies caused by incorrect user actions, invalid format or invalid input values. In these cases, the system must issue the appropriate messages to the user, and then return to the operational state that preceded the invalid (inadmissible) command or the incorrect data entry. Screen forms should be designed taking into account the requirements of unification:

All the screen forms of the user interface must be executed in a single graphic design, with the same arrangement of the main controls and navigation;similar symbols, buttons and other control (navigation) elements should be used to indicate similar operations. The terms used to denote typical operations (adding an information entity, editing the data field), as well as the sequence of user actions when executing them, must be unified;

The external behavior of similar interface elements (the response to mouse pointer hovering, focus switching, button pressing) should be implemented identically for the same type of elements. The system must meet the requirements of ergonomics provided that it is equipped with high-quality equipment (PC, monitor and other equipment).

# 3. STUDENT HOUSING SEARCH SYSTEM DEVELOPMENT AND IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

## 3.1Programming Tools Used

After analyzing the subject area and the structure of the system, the least expensive software to consider in creating the software is as follow:PHP, MYSQL JAVA Script, HTML, JQUERY, CSS.

3.1.1 Hypertext Preprocessor (PHP)

PHP is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions <?php and ?> that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

3.1.2Hyper Text Markup Language (HTML)

First developed by Tim Berners-Lee in 1990, HTML is short for HyperText Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

HTML is written in the form of HTML elements consisting of tags, enclosed in angle brackets (like <html>), within the web page content. HTML tags normally come in pairs like <h1> and </h1>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tables, images, etc.

3.1.3Cascading Style Sheets (CSS)

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is one of the core languages of the open web and is standardized across browsers according to the W3C specification. Developed in levels, CSS1 is now obsolete, CSS2.1 is a recommendation, and CSS3, now split into smaller modules, is progressing on the standardization track.

3.1.4JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

3.1.5JQuery

JQuery is a JavaScript library that allows web developers to add extra functionality to their websites. It is open source and provided for free under the MIT license. In recent years, jQuery has become the most popular JavaScript library used in web development.

To implement jQuery, a web developer simply needs to reference the jQuery JavaScript file within the HTML of a webpage. Some websites host their own local copy of jQuery, while others simply reference the library hosted by Google or the jQuery server. For example, a webpage may load the jQuery library using the following line within the <head> section of the HTML:

*<script type="text/javascript"*

*src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>*

Once the JQuery library is loaded, a webpage can call any jQuery function supported by the library. Common examples include modifying text, processing form data, moving elements on a page, and performing animations. jQuery can also work with Ajax code and scripting languages, such as PHP and ASP to access data from a database. Since jQuery runs on the client side (rather than the web server), it can update information on a webpage in realtime, without reloading the page. A common example is "autocomplete," in which a search form automatically displays common searches as you type your query. In fact, this is how TechTerms.com provides search suggestions when you type in the search box.

3.1.6MySQL

MySQL is the world’s most popular open source database, enabling the cost-effective delivery of reliable, high-performance and scalable Web-based and embedded database applications. It is an integrated transaction safe, ACID-compliant database with full commit, rollback, crash recovery, and row-level locking capabilities. MySQL delivers the ease of use, scalability, and high performance, as well as a full suit of database drivers and visual tools to help developers and DBAs build and manage their business-critical MySQL applications. MySQL is developed, distributed, and supported by Oracle, and the latest information about MySQL software can be found on the MySQL Web site. The MySQL database provides the following features:

* high performance and scalability to meet the demands of exponentially growing data loads and users;
* self-healing replication clusters to improve scalability, performance and availability;
* online schema change to meet changing business requirements.
* performance schema for monitoring user- and application-level performance and resource consumption;
* sql and nosql access for performing complex queries and simple, fast key value operations;
* platform independence giving you the flexibility to develop and deploy on multiple operating systems;
* big data interoperability using mysql as the operational data store for hadoop and cassandra.

To achieve the highest level of MySQL scalability, security, reliability and uptime, MySQL Enterprise Edition includes the most comprehensive set of advanced features, management tools and technical support, including MySQL Enterprise Monitor, MySQL Enterprise Backup, as well as scalability, security, auditing and high availability features. MySQL Enterprise Edition also includes 24x7 Oracle Premier Support and Oracle product certifications and integrations.

## 3.2 Software Structure (classes, modules, subroutines)

3.2.1System Modules

The software for the automation of the student housing search system includes the following modules that run on the server and perform their specific function:

about.php−this module display about the hotel information

add.php− this module is about adding housing details on the website;

connect\_database.php− this module connect the backend and the frontend for easy access of information on the website;

contact.php−this module contains the page where the customer can contact the

housing owner by sending message;

details.php− this module is about full information about the housing;

edit.php− this module is about editing the housing information;

favorite.php−this module helps the admin to add city to the database in the

admin panel;

footer.php− this module is gives information about external links to other profiles or websites;

head.php−this module contains the head section of the website;

housing.sql− this module is about creating the tables in the database;

index.php−opening the main page;

login.php−this module gives the users to login to the system;

logout.php−this module logs the users out of the system;

messages.php−this module enables the admin to check for messages;

myupload.php− this module enables the admin/housing owners to check for uploads;

search.php−this module handles the search field, where the customer can find

region and city in the system;

signup.php− this module enables the admin to signup on the website;

## 3.3 Code Description

Below there are the description of some important functions of the system.

Table 2.7 – Description of codes

|  |  |
| --- | --- |
| Codes | Description |
| *mysqli\_connect()* | Open a connection to a MySQL server |
| *Session\_start()* | Start a new or resume existing sission |
| *isset()* | Determine if a variable is set and is not null |
| *mysqli\_query()* | Performs a query on the database |
| *header()* | Send a raw HTTP header |
| *require\_once()* | PHP will check if the file has already been included, and if so, do not include(require) it again. |
| *mysqli\_num\_rows()* | Gets the number of rows in a result |
| *$\_GET[]* | HTTP GET variables |
| *mysqli\_fetch\_array()* | Fetch a result as an associative array, or both |
| *empty()* | Determine whether a variable is empty |
| *move\_uploaded\_file()* | Moves an uploaded file to a new location |
| *mysqli\_insert\_id()* | Get the id generated in the last query. |
| *strtolower()* | Make a string lowercase. |
| *pathinfo()* | Returns information about a file path. |
| *unset()* | Unset a given variable. |
| *Mail* | Sends an email. |
| *in\_array()* | Check if a value exists in an array |
| *Connect\_database.php*  *<?php*  *$connect = mysqli\_connect("localhost", "root", "", "housing");*  *if (!$connect) {*  *die("Connection failed: " . mysqli\_connect\_error());}?>* | 1. Declaring the connection variable for database connection.  2. Setting a rule: If connection with database fails, and error occurs. |
| Continuation of Table 2.7 | |
| *About.php*  *<?php*  *session\_start();*  *$page = 'about';*  *?>*  *<!DOCTYPE html>*  *<html lang="en">*  *<head>*  *<title>About</title>*  *<?php include('head.php') ?>*  *<style type="text/css">*  *body{background-color: #F7F9F9;}*  *.card:hover{ box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);}*  *.act{background-color: #D5DBDB;border-radius: 25px;}*  *a{color: black;}*  *.heart:hover{color:red !important;}*  *</style>*  *</head>*  *<body>*  *<?php include('nav.php') ?>*  *<?php include('footer.php') ?>*  *</body>*  *</html>* | 1. Declaring variable for start [page = ‘about’] while initiating session\_start().  2. Styling page using the style tag (for tag body - a, class card:hover – act – heart:hover ).  3. The include() function takes all the text in a specified file and copies it into the file that uses the include function specified file such as, ( nav.php, footer.php, head.php ). |
| *<?php*  *session\_start();*  *require\_once('connect\_database.php');*  *$page = 'messages';*  *$select = "SELECT \*FROM messages where user\_id = '".$\_SESSION['user\_id']."'";*  *$result = mysqli\_query($connect, $select);* | Declaration: a variable $page and setting value as messages. $select variable used as query to call from messages database. $result variable to connect to the query $select. |
| *if (isset($\_GET['delete'])) {*  *$msg\_id = $\_GET['delete'];*  *$del\_msg = "DELETE FROM messages WHERE msg\_id = '$msg\_id'";*  *$del\_res = mysqli\_query($connect, $del\_msg);* | Setting rule [if statement]: if delete is initialized, msg id will be called using $del\_msg variable which is used as a query to call the messages database. |
| *<?php }*  *} else{ echo '<center><h5>Empty</h5></center>';}?>* | Rule [if statement]: If rows for result aren’t found and less than 0, echo a message Empty. |
| *if ($del\_res) {*  *header('location:messages.php');*  *}}?>*  *<!DOCTYPE html>*  *<html lang="en">*  *<head>*  *<title>Messages</title>*  *<?php include('head.php') ?>*  *<style type="text/css">*  *body{background-color: #F7F9F9;}*  *.act{background-color: #D5DBDB;border-radius: 25px;}*  *a{color: black;}*  *</style>*  *</head>*  *<body>*  *<?php include('nav.php') ?>*  *<div class="container" style="margin-top: 50px;">*  *<div class="row">*  *<div class="col-3"></div>*  *<div class="col-6">*  *<?php*  *if(mysqli\_num\_rows($result) > 0)*  *{*  *while ($row = mysqli\_fetch\_array($result)) {*  *?>*  *<div class="card container" style="margin-top: 5px;"><a style="text-align: right;" href="messages.php?delete=<?php echo $row['msg\_id'] ?>" onclick="return confirm('Are you sure to delete this Message?')"><i class="fas fa-times"></i></a>*  *<h6 class="card-title"><?php echo $row['sender'].", ".$row['sender\_email']; ?></h6>*  *<p class="card-text"><?php echo $row['message']; ?></p>*  *<p class="card-text"><small class="text-muted"><?php echo date("d.m.Y", strtotime($row['msg\_date']))." At ".date("H:i", strtotime($row['msg\_time']))?></small></p>*  *</div>* | Rule [if statement]: If rows for result are more than 0, fetch the result from the array (database) |
| *</div>*  *<div class="col-3"></div>*  *</div>*  *</div>*  *<?php include('footer.php') ?>*  *</body>*  *</html>* | The include() function takes all the text in a specified file and copies it into the file that uses the include function specified file such as, ( nav.php, footer.php, head.php ). |
| ***search.php***  *<?php*  *session\_start();*  *require\_once('connect\_database.php');*  *if (isset($\_POST['search'])) {*  *$min = $\_POST['min'];// minimum price*  *$max = $\_POST['max']; // maximum price*  *$num = $\_POST['num']; //number of rooms*  *$city = $\_POST['city'];* | Rule: declaring min, max, num, city variable, if the search is found/ not found that variable will display min, max, num, city. |
| *$select = "SELECT \*FROM apartments*  *INNER JOIN images on apartments.apartment\_id = images.apartment\_id*  *WHERE*  *apartments.city = '$city' AND*  *apartments.num = '$num' AND*  *apartments.price BETWEEN '$min' AND '$max'*  *GROUP BY apartments.apartment\_id*  *ORDER BY apartments.date\_upload, apartments.time\_upload DESC";*  *$result = mysqli\_query($connect, $select);*  *}* | Declare $select to be used as a query to select from apartments database. |
| *?>*  *<!DOCTYPE html>*  *<html lang="en">*  *<head>*  *<title>Search</title>*  *<?php include('head.php') ?>*  *<style type="text/css">*  *body{background-color: #F7F9F9;}*  *.card:hover{ box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);}*  *.act{background-color: #D5DBDB;border-radius: 25px;}*  *a{color: black;}*  *</style>* | Using the style tag , we design the outward look of our page. |
| *</head>*  *<body>*  *<?php include('nav.php') ?>*  *<br/><br/><br/>*  *<div class="container" style="margin-top: 50px;">*  *<?php*  *if(mysqli\_num\_rows($result) > 0)*  *{*  *echo count(mysqli\_num\_rows($result))." Result(s) found";*  *while ($row = mysqli\_fetch\_array($result)) { ?>* | Rule: if the number of result found is more than 0 , count the number of result and while doing that fetch the search name by include it in an array. |
| *<div class="card container" style="margin-top: 20px;">*  *<div class="row">*  *<div class="col">*  *<img src="<?php echo $row['img\_path']; ?>" class="img-thumbnail" alt="photo">*  *</div>*  *<div class="col">*  *<?php echo $row['city'].", ".$row['region'].", ".$row['street']; ?> <br/>*  *<h5><?php echo $row['price']?> $/mo\*</h5>*  *<h6>Number of rooms: <?php echo $row['num']?></h6>*  *<p><?php echo $row['description']?></p>*  *</div>*  *<div class="col" style="text-align: right;color: gray;">*  *<p><?php echo date("d.m.Y", strtotime($row['date\_upload']))?><br/>*  *<?php echo date("H:i", strtotime($row['time\_upload']))?></p>*  *<a class="btn btn-outline-info" href="details.php?id=<?php echo $row['apartment\_id'] ?>" ><i class="fas fa-info-circle"></i> Details</a>*  *</div>*  *</div>*  *</div>*  *<?php } }else {?>*  *<center><h5><i class="far fa-frown"></i> No results found!<br/>search again <i class="fas fa-arrow-down"></i></h5></center>*  *<div class="bg-light" >*  *<div class="container">*  *<form action="" method="POST">*  *<div class="row">*  *<div class="col">*  *Min Price<input type="number" name="min" class="form-control" min="1" value="1" autocomplete="off">*  *</div>*  *<div class="col">*  *Max Price<input type="number" name="max" class="form-control" min="1" value="1" autocomplete="off">*  *</div>*  *<div class="col">*  *Number of rooms*  *<select name="num" class="form-control">*  *<option value="1">1</option>*  *<option value="2">2</option>*  *<option value="3">3</option>*  *<option value="4">4</option>*  *<option value="9">more</option>*  *</select>* | Using the dropdown box to selecting the number of rooms, for either 1 – 4 or more persons |
| </div>  <div class="col">  *City*  *<select name="city" class="form-control">*  *<option value="Minsk">Minsk</option>*  *<option value="Grodno">Grodno</option>*  *<option value="Brest">Brest</option>*  *<option value="Gomel">Gomel</option>*  *<option value="Mogilev">Mogilev</option>*  *<option value="Viciebsk">Viciebsk</option>*  *</select>* | Dropdown box, having the option for users to select a region to view houses using the student housing system. |
| *</div>*  *<div class="col">*  *<br/>*  *<button name="search" type="submit" class="btn btn-outline-secondary w-100"><i class="fas fa-search"></i> Search</button>*    *</div>*  *</div>*  *</form>* | Using the form tag to callup criteria already set within the form e.g.min-price, max-price, number of rooms, city. |
| *</div>*  *</div>*  *<?php } ?>*  *</div>*  *<?php include('footer.php') ?>*  *</body>*  *</html>* | 1. The include() function takes all the text in a specified file and copies it into the file that uses the include function specified file such as, ( nav.php, footer.php, head.php ). |

## Implementation of the User Manual

3.4.1Main Page

On the main page, the user(student) can search for housing from the search bar on the top, under the navigation bar, he/she can enter the following details for the search: Min Price: is the minimum price of housing he/she is looking for; - Max Price: is the maximum price of the housing; Rooms: number of rooms in the housing; City: is the city where the housing is located.

The user can also browse the main page and check all housing on the website under the search bar, which are ordered by date and time of upload, or he/she can select the city from the ‘Search by city’ bar on the left side of the page under the search bar, which will show all the housing in the chosen city.

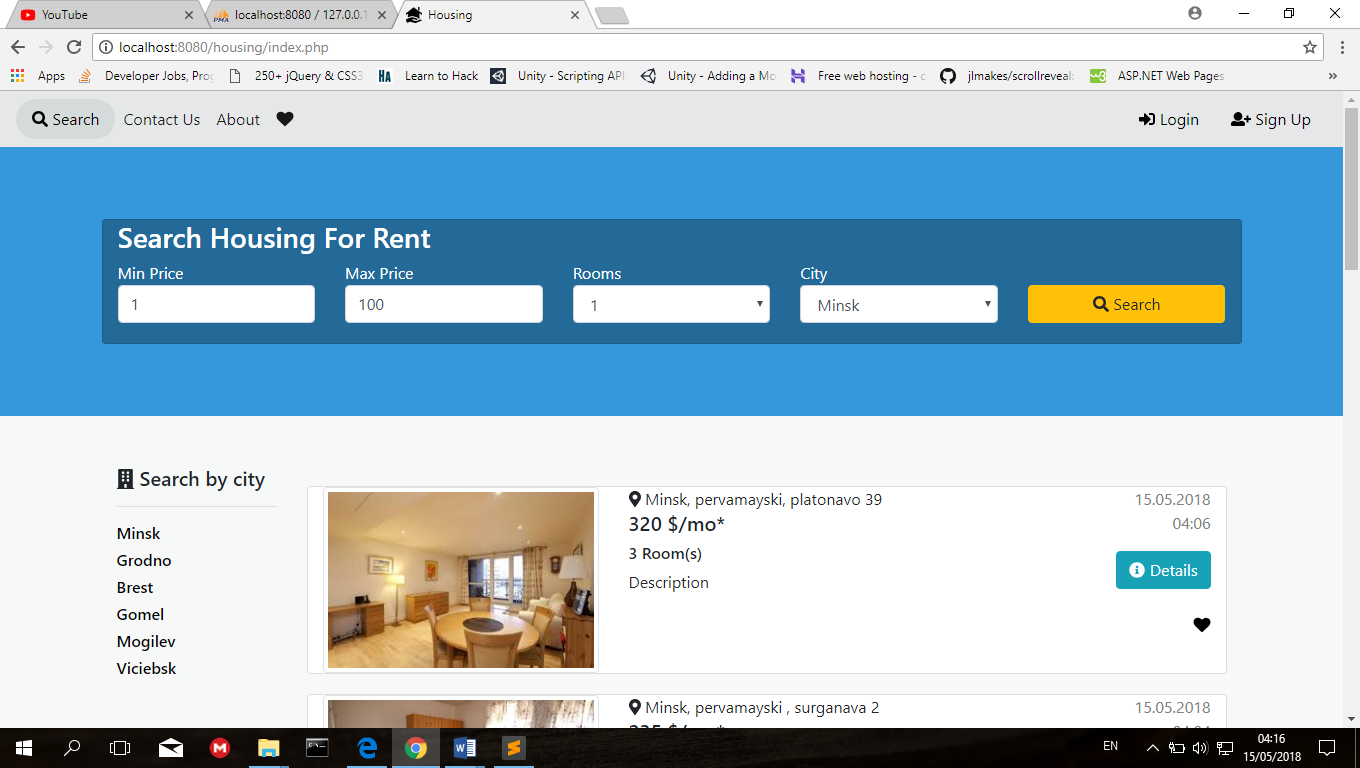


Figure 3.1 – Main page

After submitting search details, it will redirect the user to search results page, where the user can see the housing to his specific demands, in case no results were founded, it will appear a message “No Results found” and under it a search bar for him/her to search again, if results were found it will appear the number of results found.

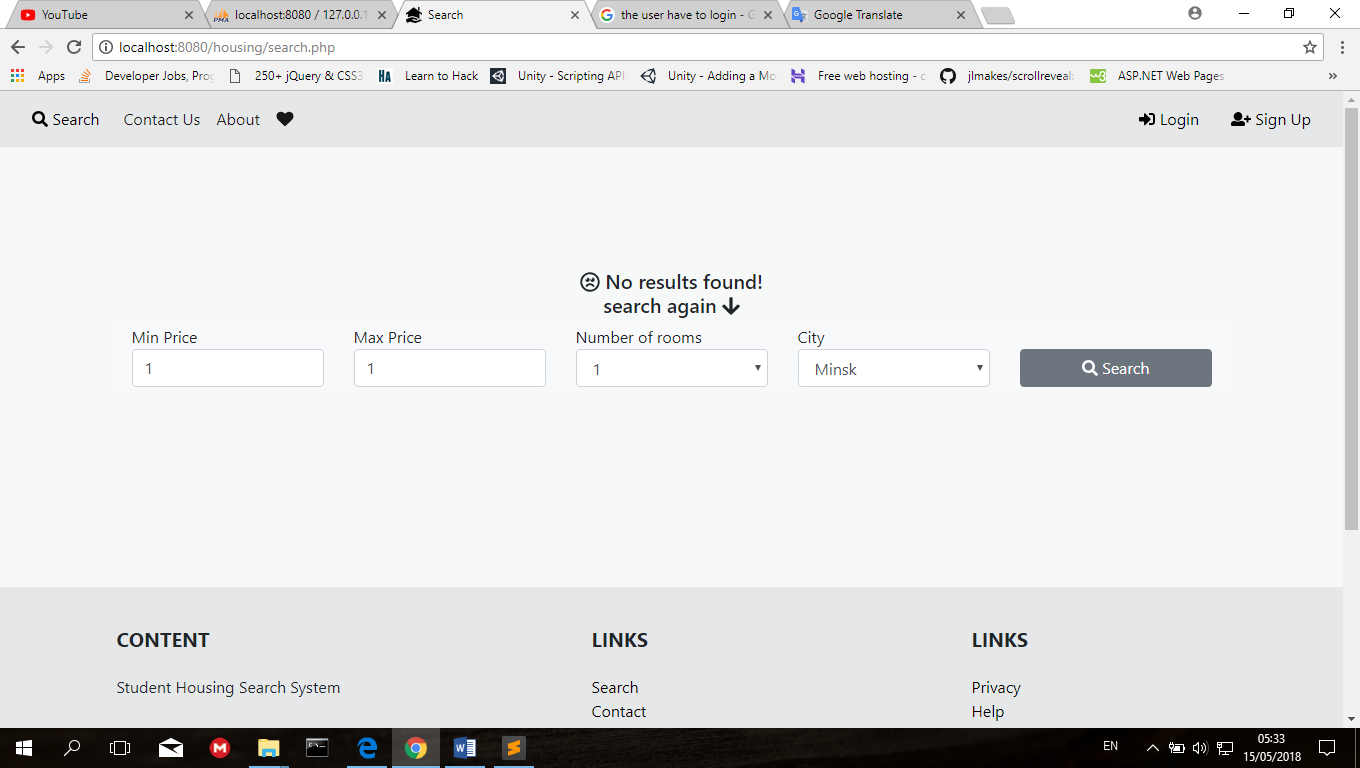


Figure 3.2 – No results found.

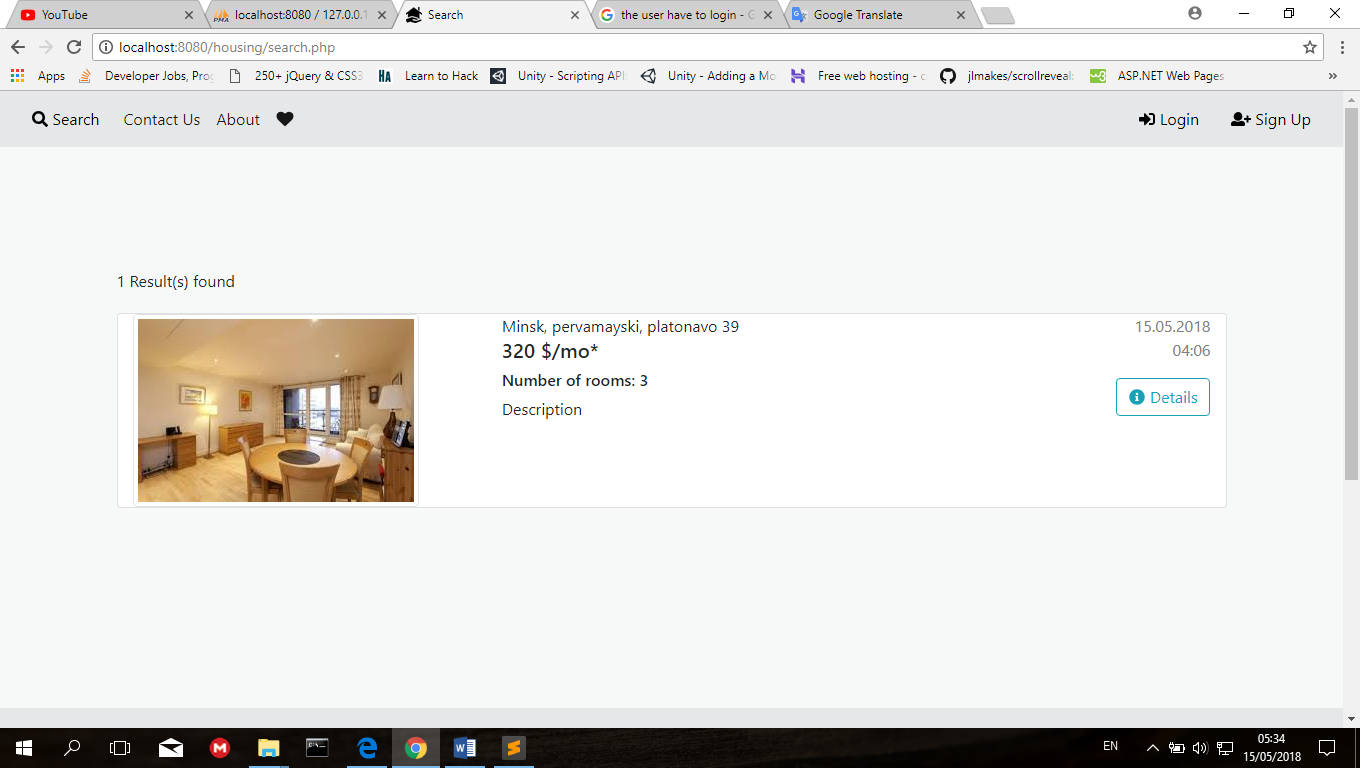


Figure 3.3 – Results found

3.4.2Details Page

For each housing there is a ‘Details’ button, when the user presses this button it will redirect the user to the details page of the housing, it will show him/her a slide-show for the images of the housing where there are all uploaded images.

Under the slide-show, is the housing details and contact information, after the housing details, is a contact form where the user can send messages for the owner of the housing, in case he/she wants to ask something about it.

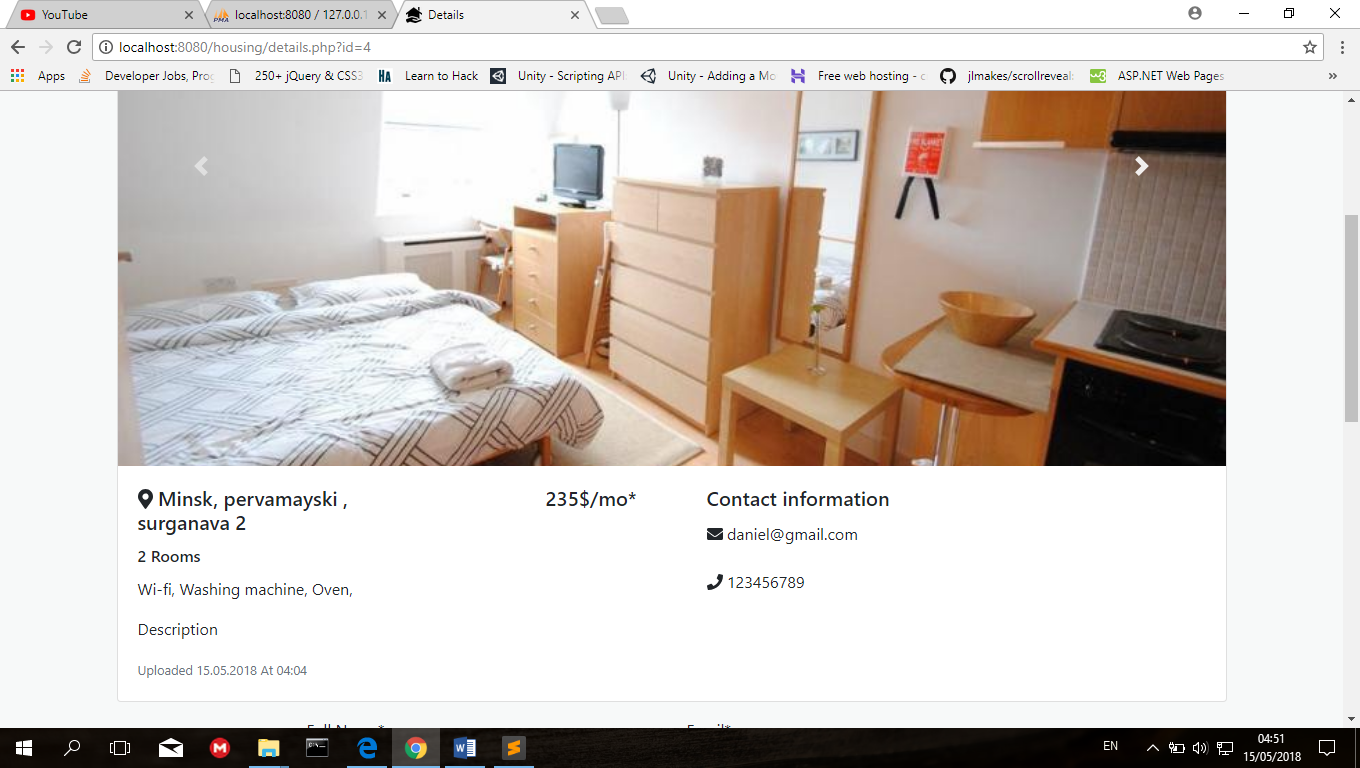


Figure 3.4 – Details page

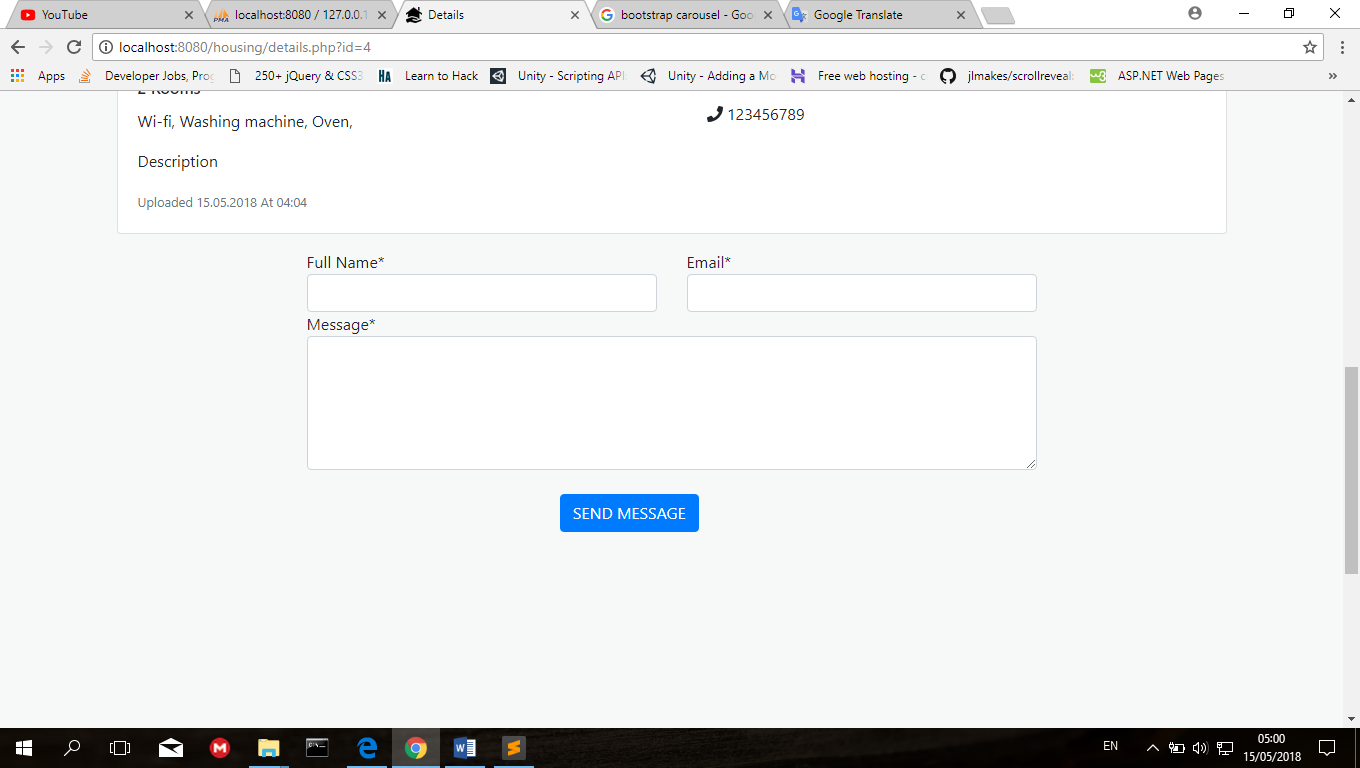


Figure 3.5 – Contact form

3.4.3Favorites Page:

For each housing there is a ‘heart ♥’ icon, when the user presses this icon it will add this housing to the favorits page, for the user to check the housing later while browsing.

The user can access the favorites page from the navigation bar by pressing on the ‘heart’ icon, it will also show him/her the number of added housings.

The user can remove the housing from favorites page, by pressing on the ‘x’ icon on the top right of the housing card.

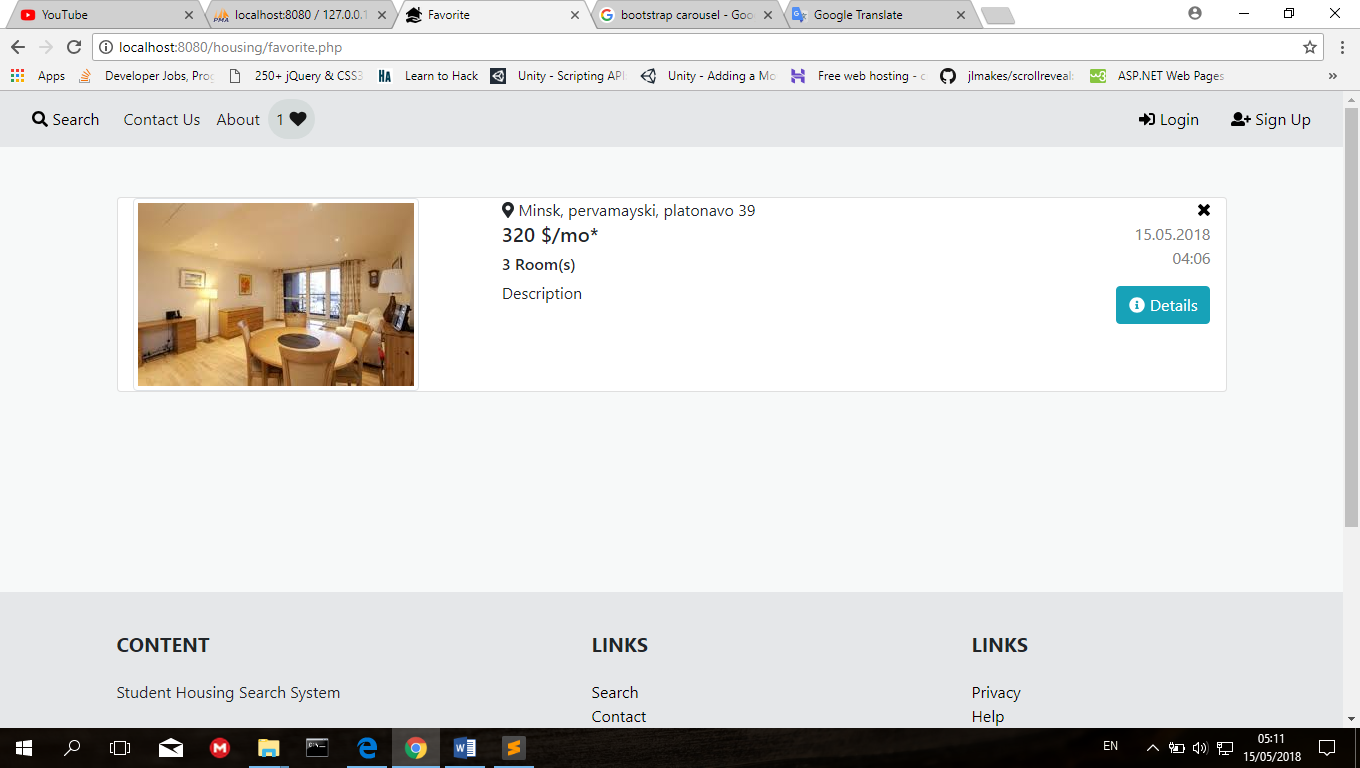


Figure 3.6 –Favorites page

3.4.4Login/Sign up Page

In order to upload housings, the owner has to sign up to the website, owner has to enter the following details to sign up: first name, last name, email, phone number, password.

Email and password are to login after signing up.

Email and phone number are to be showed in the contact information of the housing details page.

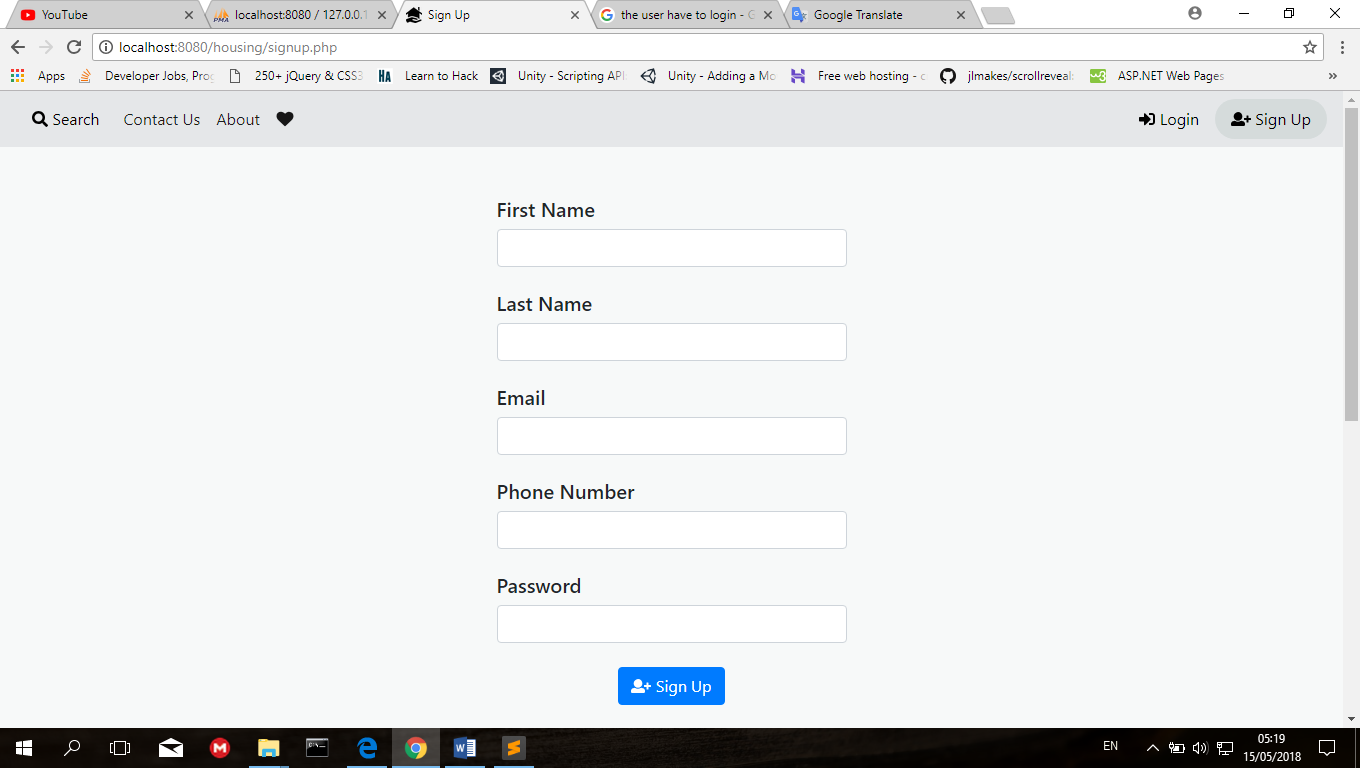


Figure 3.7 –Sign up form

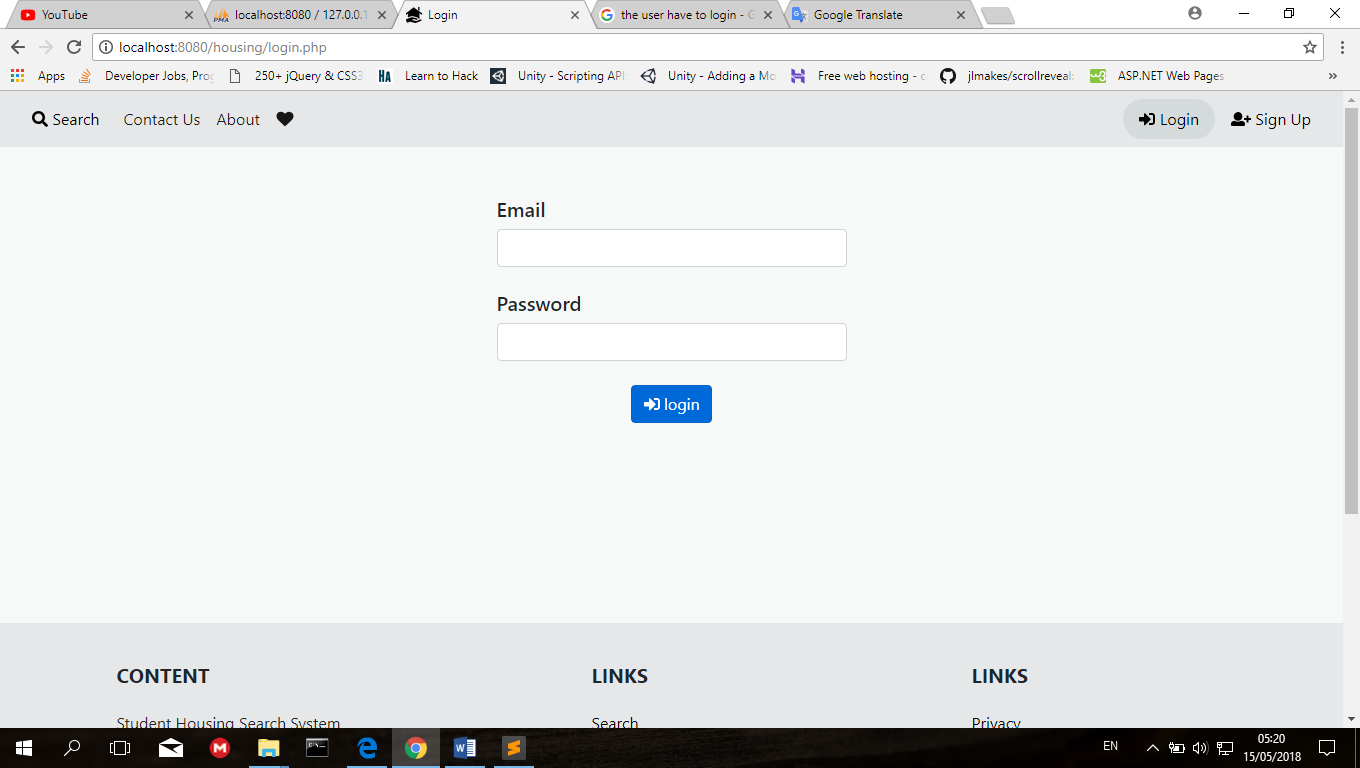


Figure 3.8 –Login form

3.4.5Add Housing Page:

After logging in the onwer can upload housing from the link ‘Add Housing’ on the navigation bar on top.

Then will enter the housing details from the form and add images of the housing, can be multiple images added at once.

After submitting the form, the housing will be showed on the main page for users(Students) to see.

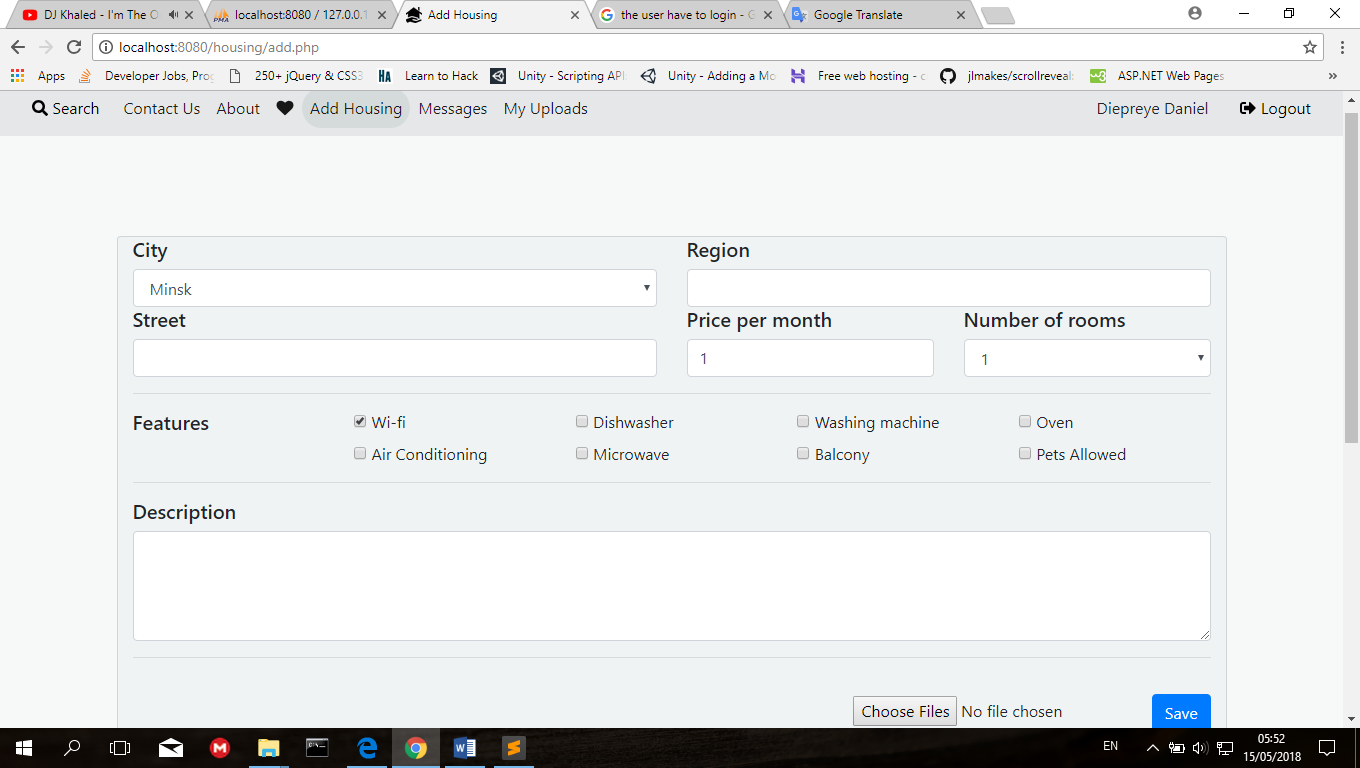


Figure 3.9 –Add housing page

3.4.6Messages Page

The owner can check messages from the ‘Messages’ link on the navigation bar on top, it will show messages sent by users(students) from the message form on the housing details page, the owner can reply to the user by email, which is showen on message card on top, it shows also time and date of the message when it was sent.

The owner can delete the message by pressing on the ‘x’ icon on the top right of the message.

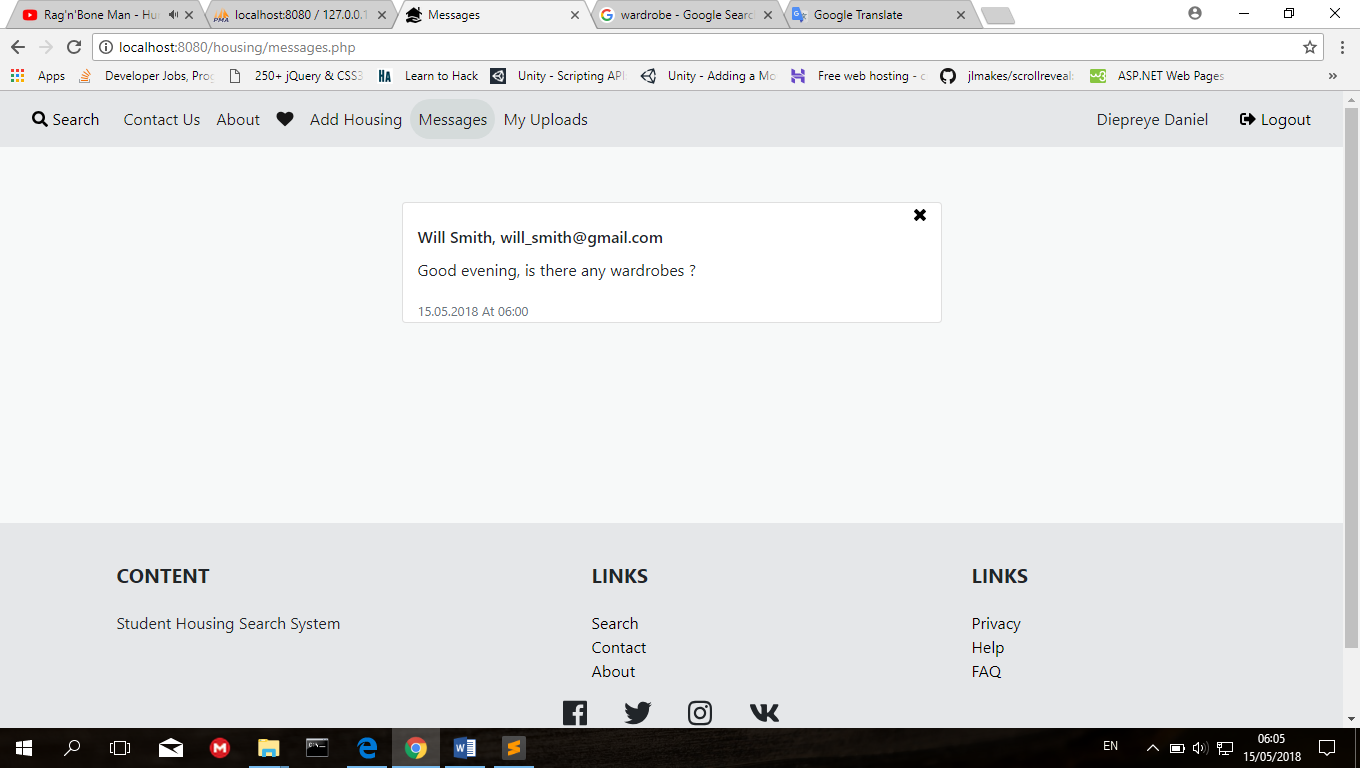


Figure 3.10 –Messages page

3.4.7Uploads Page

The owner can check uoloads from the ‘My Uploads’ link on the navigation bar on top, On this page the onwer can check the housings he/she uploaded, also can delete or edit the housing.Deleting is by pressing the red ‘Delete’ button on the right.

Editing is by pressing the green ‘Edit’ button on the right before the delete button.

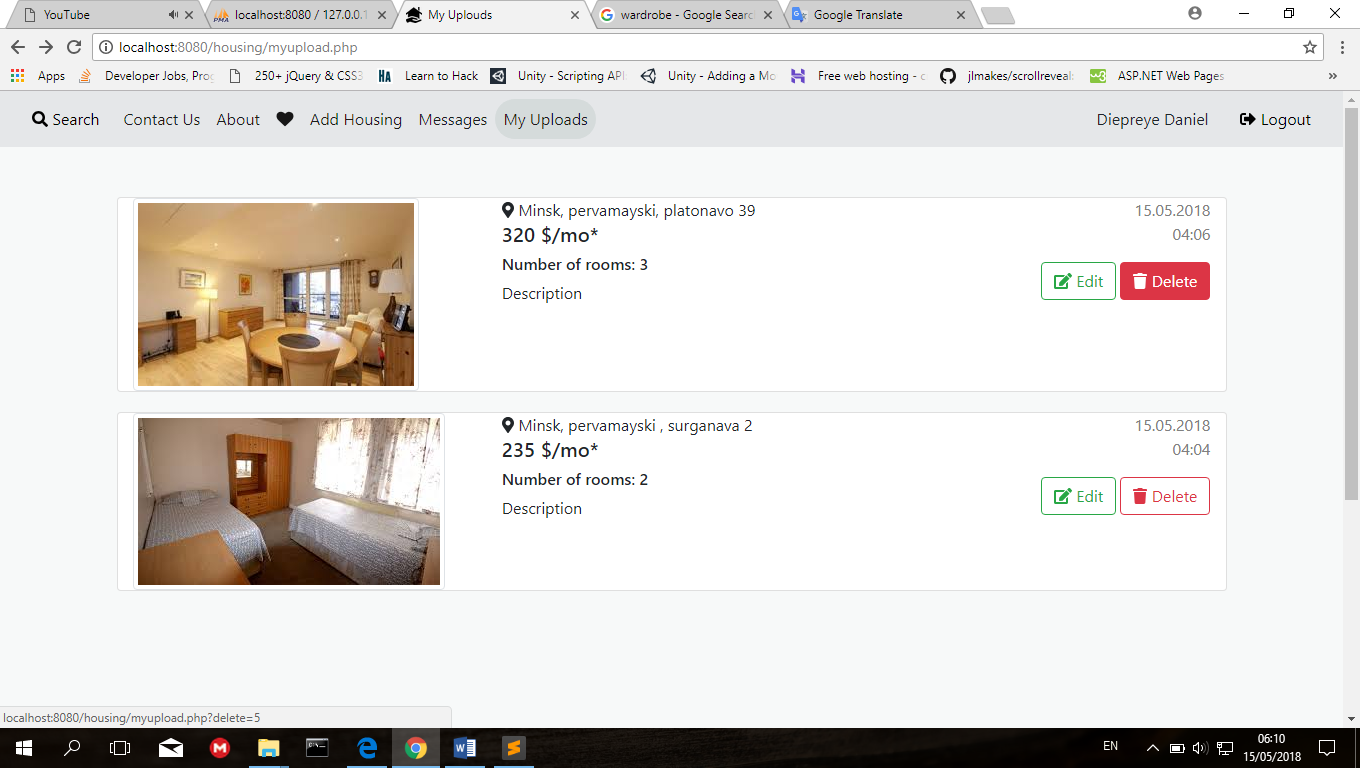


Figure 3.11 –Uploads page

3.4.8Edit Upload Page

Editing includes changing details about the housing in case there was a mistake while uploading, or to add new details to the housing, it can also delete fetures and images or add new fetures and images, also changing location details, price, number of rooms and description.

For each imagethere is a delete button where the owner can delete the image.

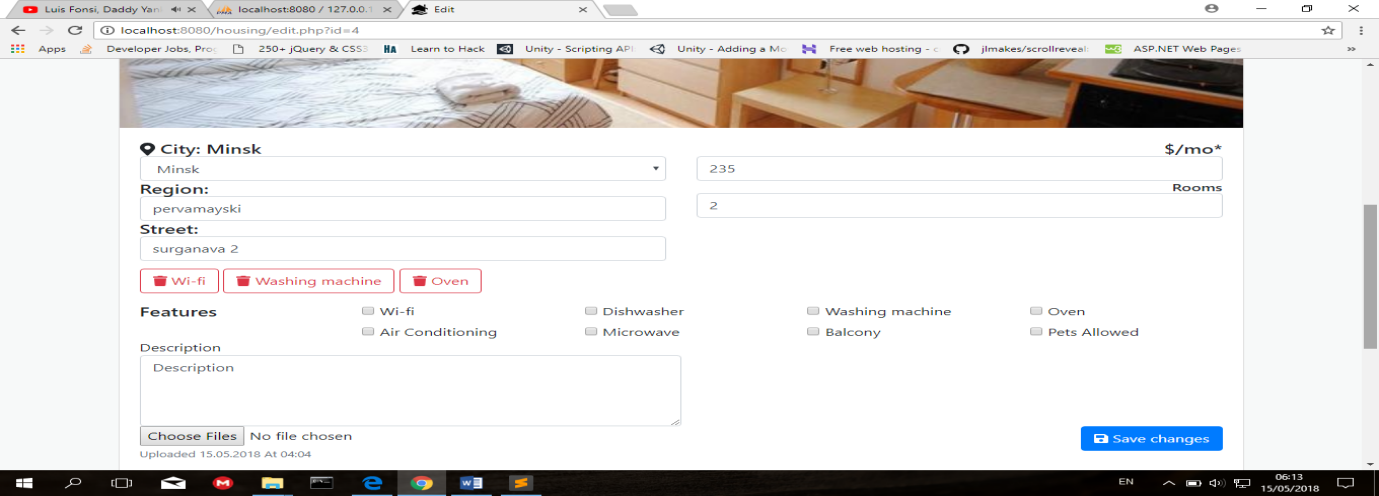


Figure 3.12–Edit Upload page

# 4 ECONOMIC FEASIBILITY STUDY

## 4.1 Characteristic of an economic case of the project

The diploma project is mainly aimed at developing a web-based application to help users/students get houses in accordance with their information, which is very important when searching for accommodation. This system will be useful for students from different countries searching for student housing.

## 4.2 Calculation of cost of materials for project accomplishment

The estimate of costs for carrying out of scientifically research work settles payments under following clauses. Calculation is performed under the formula:

(4.1)

Where; K*TP* – the coefficient considering hauling expenses

(K*TP*from 1.0 to 1.10) for the project we accept K*TP =* 1*;*

H*pi –* norm of the expenses a material kind on the project;

C*i* – unit of selling price of material kind, ruble;

N – Quantity of applied kinds of materials.

Table 4.1 - Calculation of costs for materials

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | The name of materials | Unit of measure | The price, ruble. | Quantity | The sum, ruble. |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Paper format А1 | Sheet | 3 | 8 | 18 |
| 2 | Paper format А4 | Sheet | 0.08 | 200 | 6.8 |
| 3 | Stationery | - | - | - | 60 |
| 4 | Materials for experiences and designing | - | - | - | 55 |
| The sum of expenses | | - | - | - | 145.8 |

The estimate of costs for carrying out of scientifically research work settles payments under following clauses:

## 4.3 Calculation of a base salary of the personnel occupied with accomplishment of works under the project.

The size of costs settles payments under the formula:

(4.2)

Where; T*ci* – a wage rate for a day, categories of workers, ruble;

N*i* – quantity of workers of a category;

t*i –* time of actual work of the worker of a category under the project, day;

K*np -* coefficient of awards on bonus systems

(K*np* from 1.10 to 1.40) for the project we accept K*np = 1.2;*

Calculation of the produce in the table:

Table 4.2- Base salary calculation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № п/п | The name of categories of workers and posts | Quantity of units, the people | Salary for one month, ruble. | Coefficient of bonus surcharges | Expend-itures of labour, months | The sum, ruble. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | The supervisor of studies of the project | 1 | 800 | 1.8 | 3 | 3500 |
| 2 | The engineer | 2 | 500 | 1.5 | 3 | 6250 |
| The sum of expenses | | - | - | - | - | 9750 |

## 4.4 Calculation of an additional salary of the contractors, including the various payments

Provided by the labour law, under the formula:

Additional wages include a variety of performers stipulated by the labour legislation of the payment and is calculated according to the formula:

, (4.3)

Where, Hnw – the specification of an additional salary;

Hnw from 10 to 25%, for the project it is accepted Hnw = 20%.

= 1950 rubles.

## 4.5Calculation of deductions to social insurance under the formula:

, (4.4)

Where, Hoc – rate of deductions on social ensurance (tax), Hoc = 34.6%

= ruble

## 4**.6Calculation of expenses on scientific business trip under the formula:**

We calculate the other expenses for materials scientific and technical information and the fee for the use of internet and telephone, etc.

The cost is calculated according to the formula:

, (4.5)

Where, H*kom* – the specification on scientific business trip expenses,

H*kom* from 5 to 20%, for the project we accept H*kom* = 20%.

= 9750 = ruble

## 4.7Calculation of common enterprise expenses under the formula:

Indirect cost includes the cost of management and overhead cost, calculated according to the formula”

, (4.6)

Where, Нkос - the specification of indirect expenses, Нkос ≈ from 50 to 100 %, for the project it is accepted Нkос = 90 %

= 9750 = 8,775 ruble

## 4.8Calculation of the complete cost value of the project:

The total cost of scientific and technical products is determined as the sum of all cost in all respects (clauses 1-6) as according to the formula:

Cn = Pm + Pow + Pnw + Poc + Pkom + Pkoc(4.7)

Cn = 145.8+9750+1950+ ruble

## 4.9On level of profitability in percentage of the complete cost value the profit settles payments:

At the average level of profitability in percent of the total cost is determined by the target profit unit of scientific and technical products according to the formula:

Pr = (4.8)

Where, Yp - profitability level, Yp ≈ from 10 to 30 %, for the project we accept Yp = 30 %.

Pr =⋅ = 7,985.7 ruble

## 4.10Calculation of the price of the project under the formula:

To determine an approximate (estimated) wholesale price of scientific and technical products according to the formula,

Bn = (4.9)

Bn = +7,985.7= 3 ruble

## 4.11Calculation of the tax to value added (VAT) under the formula:

The Value Added Tax is determined by the formula:

VAT = (4.10)

Where, Hvat - the tax rate on vat (the tax), Hvat = 20%.

VAT = 3 = 6,902.94 ruble

**4.11 Calculation of the price of the project taking into account the VAT under the formula:**

To determine the selling price of scientific and technical products with VAT according to the formula:

B= Bn +VAT (4.11)

B = 3+ 6,902.94= 41,525.64 ruble

Calculation of costs for the project and the project price are resulted in   
table 3.

Table 4.3 − The Estimate of costs for the project

|  |  |  |  |
| --- | --- | --- | --- |
| 1№ | Clauses of costs | Calculation | The sum, ruble. |
|  | 2 | 3 | 4 |
| 1 | Materials () | Table 1 | 145.8 |
| 2 | Base salary () | Table 2 | 9750 |
| 3 | The additional salary () |  | 1950 |
| 4 | Deductions in population social insurance fund () |  |  |
| 5 | Scientific business tripexpenses () |  | 1950 |
| 6 | Common enterprise expenses(Ркос) | 9750 | 8,775 |
| 7 | Total the cost value (Cn) | 145.8+9750+1950+ | 26,619 |
| 8 | Profit () |  | 7,985.7 |
| 9 | The project price () | +7,985.7 | 7,985.7 |
| 10 | The value-added tax () | 3⋅ | 6,902.94 |
| 11 | The price from the VAT () | 3+ 6,902.94 | 41,525.64 |

**Conclusions**: The developed system provides more help to different students searching for accommodation, cost for the student housing search system development have constituted 41,525.64 rubles.

# ConclusIon

This project aimed at developing system to replace the manual process of housing system. Where, Customer can have access to the housing by face to face before making reservation. Even if the customer has the contact of the housing to make reservation by calling, it is not guaranteed. Also, if the customer visit the housing to make reservation, there is no good database management system to keep the records of customers. The system does not serve the customer in a better way; rather it makes customer data vulnerable, because records are kept in paper

This Online Student Housing Search System serves customers well. Keeping proper records of customers for emergency and security purposes. The housing advertising effort is now accompanied by a virtual tour created on the system. Customers can have access to housing information from the system and make reservation from anywhere around the world. And after making reservation, customer will receive an email alert. Admin can also make changes in the database and view housing details and customer details as well.

# References

[1] Ramez Elmasri, Shamkant B. Navathe / Fundamentals of database systems 4th ed.

[2] Freeman Adam Pro ASP.NET MVC 5 Framework / A. Freeman. – Apress, 2013. – 784 p.

[3] Joe Mayo / Microsoft Visual Studio 2010 A Beginner’s Guide.

[4] <http://www.ijiset.com/v1s9/IJISET_V1_I9_92.pdf>.

[5] N. Sfetcu, Web Design & Development, 2014.

[6]M. Taylor, A General Understanding of Microsoft Excel, Visual Studio and Webmatrix2, Mark Taylor and Paragon Publishing, Rothersthorpe, 2013.

[7] <http://agile.csc.ncsu.edu/SEMaterials/UMLOverview.pdf>.

[8]http://www.dfit.dfinalsolution.com/dotnet%20tutorial%20for%20beginners.pdf.

[9] http://eloquentjavascript.net/Eloquent\_JavaScript.pdf

[10] <http://www.tutorialspoint.com/javascript/javascript_tutorial.pdf>

[11] Andrei Brakovich, Database and data banks (2015-2016), Lecture Materials. Belarusian State University of Informatics and Radioelectronics

Cumming, AGentleIntroductionto SQL(2003), An interactive tutorial.

[12]School of Computing of Napier University, Edinburg, UK, available at *http://sqlzoo.net* , pp 320-325.

[13] V. DeBolt, Mastering Integrating Html and CSS.

[14] H. Bidgoli, Essentials of Software Engineering 2011.

# Appendix A

**(mandatory)**

**System listing**

**A.1 Login**

*<?php*

*session\_start();*

*require\_once('connect\_database.php');*

*$page = 'login';*

*if (isset($\_POST['login'])) {*

*$email = $\_POST['email'];*

*$password = $\_POST['password'];*

*$query = "SELECT \*FROM users where user\_email = '$email' AND user\_psw = '$password'";*

*$result = mysqli\_query($connect, $query);*

*$user = $result->fetch\_assoc();*

*if( $result->num\_rows > 0) {*

*$\_SESSION['user\_logged\_in'] = true;*

*$\_SESSION['user\_fname'] = $user['user\_fname'];*

*$\_SESSION['user\_lname'] = $user['user\_lname'];*

*$\_SESSION['user\_id'] = $user['user\_id'];*

*header("refresh:0.5;url=index.php");*

*}else*

*{*

*$error = '<i class="fa fa-exclamation-triangle"></i> User dosent Exist!';*

*}*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<title>Login</title>*

*<?php include('head.php') ?>*

*<style type="text/css">*

*.act{background-color: #D5DBDB;border-radius: 25px;}*

*a{color: black;}*

*body{background-color: #F7F9F9;}*

*</style>*

*</head>*

*<body>*

*<?php include('nav.php') ?>*

*<form action="" method="POST">*

*<div class="container" style="margin-top: 50px;">*

*<div class="row" >*

*<div class="col"></div>*

*<div class="col">*

*<?php*

*if (isset($error)) {?>*

*<p style="color: red;"><?php echo $error; ?></p><br/>*

*<?php*

*}*

*?>*

*<h5>Email</h5>*

*<input type="email" name="email" class="form-control" required/><br/>*

*<h5>Password</h5>*

*<input type="password" name="password" class="form-control" required autocomplete="off"/><br/>*

*<center>*

*<button type="submit" name="login" class="btn btn-outline-primary"><i class="fas fa-sign-in-alt"></i> login</button>*

*</center>*

*</div>*

*<div class="col"></div>*

*</div>*

*</div>*

*</form>*

*<?php include('footer.php') ?>*

*</body>*

*</html>*

**A.2 SignUp**

*<?php*

*session\_start();*

*require\_once('connect\_database.php');*

*$page = 'signup';*

*if (isset($\_POST['signup'])) {*

*$fname = $\_POST['fname'];*

*$lname = $\_POST['lname'];*

*$email = $\_POST['email'];*

*$phone = $\_POST['phone'];*

*$password = $\_POST['password'];*

*$result = $connect->query("SELECT \*FROM users WHERE user\_email='$email'") or die($connect->error());*

*if ( $result->num\_rows > 0 ) {*

*$error = '<i class="fa fa-exclamation-triangle"></i> User already exist!';*

*}else*

*{*

*$sql = "INSERT INTO users(user\_fname, user\_lname, user\_email, user\_phone, user\_psw) VALUES*

*('$fname', '$lname', '$email', '$phone', '$password')";*

*$res = mysqli\_query($connect, $sql);*

*if (!$res) {*

*$error = '<i class="fa fa-exclamation-triangle"></i> Error! Could not connect to the server';*

*}else*

*{*

*header('location:login.php');*

*}*

*}*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<title>Sign Up</title>*

*<?php include('head.php') ?>*

*<style type="text/css">*

*.act{background-color: #D5DBDB;border-radius: 25px;}*

*a{color: black;}*

*body{background-color: #F7F9F9;}*

*</style>*

*</head>*

*<body>*

*<?php include('nav.php') ?>*

*<form action="" method="POST">*

*<div class="container" style="margin-top: 50px;">*

*<div class="row">*

*<div class="col"></div>*

*<div class="col">*

*<?php*

*if (isset($error)) {?>*

*<p style="color: red;"><?php echo $error; ?></p><br/>*

*<?php*

*}*

*?>*

*<h5>First Name</h5>*

*<input type="text" name="fname" class="form-control" required autocomplete="off"/><br/>*

*<h5>Last Name</h5>*

*<input type="text" name="lname" class="form-control" required autocomplete="off"/><br/>*

*<h5>Email</h5>*

*<input type="email" name="email" class="form-control" required/><br/>*

*<h5>Phone Number</h5>*

*<input type="text" name="phone" class="form-control" required autocomplete="off"/><br/>*

*<h5>Password</h5>*

*<input type="password" name="password" class="form-control" required autocomplete="off"/><br/>*

*<center>*

*<button type="submit" name="signup" class="btn btn-outline-primary"><i class="fas fa-user-plus"></i> Sign Up</button>*

*</center>*

*</div>*

*<div class="col"></div>*

*</div>*

*</div>*

*</form>*

*<?php include('footer.php') ?>*

*</body>*

*</html>*

**A.3 Housing**

*-- phpMyAdmin SQL Dump*

*-- version 4.7.4*

*-- https://www.phpmyadmin.net/*

*-- Host: 127.0.0.1*

*-- Generation Time: May 07, 2018 at 05:59 PM*

*-- Server version: 10.1.28-MariaDB*

*-- PHP Version: 7.0.25*

*SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";*

*SET AUTOCOMMIT = 0;*

*START TRANSACTION;*

*SET time\_zone = "+00:00";*

*/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;*

*/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;*

*/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;*

*/\*!40101 SET NAMES utf8mb4 \*/;*

*-- Database: `housing`*

*-- Table structure for table `apartments`*

*CREATE TABLE `apartments` (*

*`apartment\_id` int(11) NOT NULL,*

*`user\_id` int(11) NOT NULL,*

*`num` int(11) NOT NULL,*

*`city` varchar(50) NOT NULL,*

*`region` varchar(50) NOT NULL,*

*`street` varchar(50) NOT NULL,*

*`price` double NOT NULL,*

*`description` text NOT NULL,*

*`date\_upload` date NOT NULL,*

*`time\_upload` time NOT NULL*

*) ENGINE=InnoDB DEFAULT CHARSET=latin1;*

*-- Table structure for table `contact`*

*CREATE TABLE `contact` (*

*`cntc\_id` int(11) NOT NULL,*

*`cntc\_fname` varchar(50) NOT NULL,*

*`cntc\_lname` varchar(50) NOT NULL,*

*`cntc\_email` varchar(50) NOT NULL,*

*`cntc\_msg` text NOT NULL,*

*`cntc\_date\_time` datetime NOT NULL*

*) ENGINE=InnoDB DEFAULT CHARSET=latin1;*

*-- Table structure for table `features`*

*CREATE TABLE `features` (*

*`feature\_id` int(11) NOT NULL,*

*`apartment\_id` int(11) NOT NULL,*

*`feature` varchar(100) NOT NULL*

*) ENGINE=InnoDB DEFAULT CHARSET=latin1;*

*-- Table structure for table `images`*

*CREATE TABLE `images` (*

*`img\_id` int(11) NOT NULL,*

*`apartment\_id` int(11) NOT NULL,*

*`img` varchar(100) NOT NULL,*

*`img\_path` varchar(100) NOT NULL,*

*`img\_type` varchar(100) NOT NULL*

*) ENGINE=InnoDB DEFAULT CHARSET=latin1;*

*-- Table structure for table `messages`*

*CREATE TABLE `messages` (*

*`msg\_id` int(11) NOT NULL,*

*`user\_id` int(11) NOT NULL,*

*`sender` varchar(50) NOT NULL,*

*`sender\_email` varchar(50) NOT NULL,*

*`message` text NOT NULL,*

*`msg\_date` date NOT NULL,*

*`msg\_time` time NOT NULL*

*) ENGINE=InnoDB DEFAULT CHARSET=latin1;*

*-- Table structure for table `users`*

*CREATE TABLE `users` (*

*`user\_id` int(11) NOT NULL,*

*`user\_fname` varchar(50) NOT NULL,*

*`user\_lname` varchar(50) NOT NULL,*

*`user\_email` varchar(50) NOT NULL,*

*`user\_phone` varchar(50) NOT NULL,*

*`user\_psw` varchar(50) NOT NULL*

*) ENGINE=InnoDB DEFAULT CHARSET=latin1;*

*-- Indexes for dumped tables*

*-- Indexes for table `apartments`*

*ALTER TABLE `apartments`*

*ADD PRIMARY KEY (`apartment\_id`),*

*ADD KEY `user\_id` (`user\_id`);*

*-- Indexes for table `contact`*

*ALTER TABLE `contact`*

*ADD PRIMARY KEY (`cntc\_id`);*

*-- Indexes for table `features`*

*ALTER TABLE `features`*

*ADD PRIMARY KEY (`feature\_id`),*

*ADD KEY `apartment\_id` (`apartment\_id`);*

*-- Indexes for table `images`*

*ALTER TABLE `images`*

*ADD PRIMARY KEY (`img\_id`),*

*ADD KEY `apartment\_id` (`apartment\_id`);*

*-- Indexes for table `messages`*

*ALTER TABLE `messages`*

*ADD PRIMARY KEY (`msg\_id`),*

*ADD KEY `user\_id` (`user\_id`);*

*-- Indexes for table `users`*

*ALTER TABLE `users`*

*ADD PRIMARY KEY (`user\_id`);*

*-- AUTO\_INCREMENT for dumped tables*

*-- AUTO\_INCREMENT for table `apartments`*

*ALTER TABLE `apartments`*

*MODIFY `apartment\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=5;*

*-- AUTO\_INCREMENT for table `contact`*

*ALTER TABLE `contact`*

*MODIFY `cntc\_id` int(11) NOT NULL AUTO\_INCREMENT;*

*-- AUTO\_INCREMENT for table `features`*

*ALTER TABLE `features`*

*MODIFY `feature\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=12;*

*-- AUTO\_INCREMENT for table `images`*

*ALTER TABLE `images`*

*MODIFY `img\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;*

*-- AUTO\_INCREMENT for table `messages`*

*ALTER TABLE `messages`*

*MODIFY `msg\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;*

*-- AUTO\_INCREMENT for table `users`*

*ALTER TABLE `users`*

*MODIFY `user\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;*

*-- Constraints for dumped tables*

*-- Constraints for table `apartments`*

*ALTER TABLE `apartments`*

*ADD CONSTRAINT `apartments\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`);*

*-- Constraints for table `features`*

*ALTER TABLE `features`*

*ADD CONSTRAINT `features\_ibfk\_1` FOREIGN KEY (`apartment\_id`) REFERENCES `apartments` (`apartment\_id`);*

*-- Constraints for table `images`*

*ALTER TABLE `images`*

*ADD CONSTRAINT `images\_ibfk\_1` FOREIGN KEY (`apartment\_id`) REFERENCES `apartments` (`apartment\_id`);*

*-- Constraints for table `messages`*

*ALTER TABLE `messages`*

*ADD CONSTRAINT `messages\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`);*

*COMMIT;*

*/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;*

*/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;*

*/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;*

**A.4 Search**

*<?php*

*session\_start();*

*require\_once('connect\_database.php');*

*if (isset($\_POST['search'])) {*

*$min = $\_POST['min'];// minimum price*

*$max = $\_POST['max']; // maximum price*

*$num = $\_POST['num']; //number of rooms*

*$city = $\_POST['city'];*

*$select = "SELECT \*FROM apartments*

*INNER JOIN images on apartments.apartment\_id = images.apartment\_id*

*WHERE*

*apartments.city = '$city' AND*

*apartments.num = '$num' AND*

*apartments.price BETWEEN '$min' AND '$max'*

*GROUP BY apartments.apartment\_id*

*ORDER BY apartments.date\_upload, apartments.time\_upload DESC";*

*$result = mysqli\_query($connect, $select);*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<title>Search</title>*

*<?php include('head.php') ?>*

*<style type="text/css">*

*body{background-color: #F7F9F9;}*

*.card:hover{ box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);}*

*.act{background-color: #D5DBDB;border-radius: 25px;}*

*a{color: black;}*

*</style>*

*</head>*

*<body>*

*<?php include('nav.php') ?>*

*<br/><br/><br/>*

*<div class="container" style="margin-top: 50px;">*

*<?php*

*if(mysqli\_num\_rows($result) > 0)*

*{*

*echo count(mysqli\_num\_rows($result))." Result(s) found";*

*while ($row = mysqli\_fetch\_array($result)) { ?>*

*<div class="card container" style="margin-top: 20px;">*

*<div class="row">*

*<div class="col">*

*<img src="<?php echo $row['img\_path']; ?>" class="img-thumbnail" alt="photo">*

*</div>*

*<div class="col">*

*<?php echo $row['city'].", ".$row['region'].", ".$row['street']; ?> <br/>*

*<h5><?php echo $row['price']?> $/mo\*</h5>*

*<h6>Number of rooms: <?php echo $row['num']?></h6>*

*<p><?php echo $row['description']?></p>*

*</div>*

*<div class="col" style="text-align: right;color: gray;">*

*<p><?php echo date("d.m.Y", strtotime($row['date\_upload']))?><br/>*

*<?php echo date("H:i", strtotime($row['time\_upload']))?></p>*

*<a class="btn btn-outline-info" href="details.php?id=<?php echo $row['apartment\_id'] ?>" ><i class="fas fa-info-circle"></i> Details</a>*

*</div>*

*</div>*

*</div>*

*<?php } }else {?>*

*<center><h5><i class="far fa-frown"></i> No results found!<br/>search again <i class="fas fa-arrow-down"></i></h5></center>*

*<div class="bg-light" >*

*<div class="container">*

*<form action="" method="POST">*

*<div class="row">*

*<div class="col">*

*Min Price<input type="number" name="min" class="form-control" min="1" value="1" autocomplete="off">*

*</div>*

*<div class="col">*

*Max Price<input type="number" name="max" class="form-control" min="1" value="1" autocomplete="off">*

*</div>*

*<div class="col">*

*Number of rooms*

*<select name="num" class="form-control">*

*<option value="1">1</option>*

*<option value="2">2</option>*

*<option value="3">3</option>*

*<option value="4">4</option>*

*<option value="9">more</option>*

*</select>*

*</div>*

*<div class="col">*

*City*

*<select name="city" class="form-control">*

*<option value="Minsk">Minsk</option>*

*<option value="Grodno">Grodno</option>*

*<option value="Brest">Brest</option>*

*<option value="Gomel">Gomel</option>*

*<option value="Mogilev">Mogilev</option>*

*<option value="Viciebsk">Viciebsk</option>*

*</select>*

*</div>*

*<div class="col">*

*<br/>*

*<button name="search" type="submit" class="btn btn-outline-secondary w-100"><i class="fas fa-search"></i> Search</button>*

*</div>*

*</div>*

*</form>*

*</div>*

*</div>*

*<?php } ?>*

*</div>*

*<?php include('footer.php') ?>*

*</body>*

*</html>*

**A.5 Add housing**

*<?php*

*session\_start();*

*require\_once('connect\_database.php');*

*$page = 'add';*

*if (isset($\_POST['insert'])) {*

*$city = $\_POST['city'];*

*$region = $\_POST['region'];*

*$street = $\_POST['street'];*

*$num = $\_POST['num'];*

*$price = $\_POST['price'];*

*$des = $\_POST['des'];*

*date\_default\_timezone\_set('Europe/Minsk');*

*$date = date("Y-m-d");*

*$time = date("h:i:sa");*

*$query = "INSERT INTO apartments (user\_id,num, city, region, street, price, description, date\_upload, time\_upload) VALUES ('".$\_SESSION['user\_id']."','$num', '$city', '$region', '$street', '$price', '$des', '$date', '$time')";*

*$res = mysqli\_query($connect, $query) or trigger\_error(mysqli\_error(), " ".$query);*

*$apart\_id = mysqli\_insert\_id($connect);*

*if ($res) {*

*if (!empty($\_POST['feature']))*

*{*

*foreach ($\_POST['feature'] as $feature) {*

*$feature\_query ="INSERT INTO features(apartment\_id, feature) VALUES ('$apart\_id', '$feature')";*

*$feature\_res = mysqli\_query($connect, $feature\_query) or trigger\_error(mysqli\_error(), " ".$feature\_query);*

*}*

*}*

*for ($i=0; $i < count($\_FILES['myfile']['name']) ; $i++) {*

*$filetmp = $\_FILES['myfile']['tmp\_name'][$i];*

*$filename = $\_FILES['myfile']['name'][$i];*

*$filetype = $\_FILES['myfile']['type'][$i];*

*$imgExt = strtolower(pathinfo($filename, PATHINFO\_EXTENSION))[$i];*

*$pic = time().'\_'.rand(1000,9999).'.'.$imgExt;*

*$filepath = "images/".$pic;*

*move\_uploaded\_file($filetmp, $filepath);*

*$img\_query = "INSERT INTO images(apartment\_id, img, img\_path, img\_type) VALUES*

*('$apart\_id', '$filename' , '$filepath', '$filetype')";*

*$img\_res = mysqli\_query($connect, $img\_query) or trigger\_error(mysqli\_error(), " ".$img\_query);*

*}*

*echo '<script>alert("Inserted successfully")</script>';*

*}*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<title>Add Housing</title>*

*<?php include('head.php'); ?>*

*<style type="text/css">*

*body{background-color: #F7F9F9;}*

*.act{background-color: #D5DBDB;border-radius: 25px;}*

*a{color: black;}*

*</style>*

*</head>*

*<body>*

*<?php include('nav.php') ?>*

*<div class="container" style="margin-top: 100px;">*

*<form action="" method="POST" enctype="multipart/form-data">*

*<div class="card container" style="background-color: #F0F3F4;">*

*<div class="row">*

*<div class="col">*

*<h5>City</h5>*

*<select name="city" class="form-control">*

*<option value="Minsk">Minsk</option>*

*<option value="Grodno">Grodno</option>*

*<option value="Brest">Brest</option>*

*<option value="Gomel">Gomel</option>*

*<option value="Mogilev">Mogilev</option>*

*<option value="Viciebsk">Viciebsk</option>*

*</select>*

*</div>*

*<div class="col">*

*<h5>Region</h5>*

*<input type="text" name="region" class="form-control" required autocomplete="off"/>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col">*

*<h5>Street</h5>*

*<input type="text" name="street" class="form-control" required autocomplete="off"/>*

*</div>*

*<div class="col">*

*<div class="row">*

*<div class="col">*

*<h5>Price per month</h5>*

*<input type="number" name="price" class="form-control" required autocomplete="off" min="1" value="1"/>*

*</div>*

*<div class="col">*

*<h5>Number of rooms</h5>*

*<select name="num" class="form-control">*

*<option value="1">1</option>*

*<option value="2">2</option>*

*<option value="3">3</option>*

*<option value="4">4</option>*

*<option value="5">5</option>*

*<option value="6">6</option>*

*<option value="7">7</option>*

*<option value="8">8</option>*

*<option value="9">9</option>*

*</select>*

*</div>*

*</div>*

*</div>*

*</div>*

*<hr>*

*<div class="row">*

*<div class="col">*

*<h5>Features</h5>*

*</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Wi-fi" checked> Wi-fi</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Dishwasher"> Dishwasher</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Washing machine"> Washing machine</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Oven"> Oven</div>*

*</div>*

*<div class="row">*

*<div class="col">*

*</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Air Conditioning"> Air Conditioning</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Microwave"> Microwave</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Balcony"> Balcony</div>*

*<div class="col"><input type="checkbox" name="feature[]" value="Pets Allowed"> Pets Allowed</div>*

*</div>*

*<hr>*

*<div class="row">*

*<div class="col">*

*<h5>Description</h5>*

*<textarea name="des" class="form-control" rows="4"></textarea>*

*</div>*

*</div>*

*<hr>*

*<div class="row" style="margin-top: 20px;text-align: right;">*

*<div class="col">*

*<input type="file" name="myfile[]" required multiple/>*

*<button name="insert" type="submit" class="btn btn-primary">Save</button>*

*</div>*

*</div>*

*<hr>*

*</div>*

*</form>*

*</div>*

*<?php include('footer.php') ?>*

*</body>*

*</html>*

**A.6 Contact**

*<?php*

*session\_start();ousing search system. This allows students to receive information about the availability of apartments or rooms. This web application allows users to easily access all the information related to apartment rentals.*

*error\_reporting(0);*

*require\_once('connect\_database.php');*

*$page = 'contact';*

*date\_default\_timezone\_set('Europe/Minsk');*

*if (isset($\_POST['send'])) {*

*$fname = $\_POST['fname'];*

*$lname = $\_POST['lname'];*

*$email = $\_POST['email'];*

*$msg = $\_POST['msg'];*

*$date = date("Y-m-d H:i:s");*

*$sql = "INSERT INTO contact (cntc\_fname, cntc\_lname, cntc\_email, cntc\_msg, cntc\_date\_time)VALUES*

*('$fname', '$lname', '$email', '$msg', '$date')";*

*$res = mysqli\_query($connect, $sql);*

*$to = "";*

*$subject = "";*

*$from = $fname." ".$lname." ,".$email;*

*mail($to, $subject, $from, $msg);*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<title>Contact Us</title>*

*<?php include('head.php') ?>*

*<style type="text/css">*

*body{background-color: #F7F9F9;}*

*.act{background-color: #D5DBDB;border-radius: 25px;}*

*.act2{background-color: #D5DBDB;}*

*a{color: black;}*

*</style>*

*</head>*

*<body>*

*<?php include('nav.php') ?>*

*<div>*

*</div>*

*<form action="" method="POST">*

*<hr>*

*<div class="container">*

*<div class="row">*

*<div class="col-9">*

*<div class="row">*

*<div class="col-3"><h6>First Name<font color="red">\*</font></h6></div>*

*<div class="col-9"><input type="text" name="fname" class="form-control" required autocomplete="off"></div>*

*</div>*

*<br/>*

*<div class="row">*

*<div class="col-3"><h6>Last Name<font color="red">\*</font></h6></div>*

*<div class="col-9"><input type="text" name="lname" class="form-control" required autocomplete="off"></div>*

*</div>*

*<br/>*

*<div class="row">*

*<div class="col-3"><h6>Email<font color="red">\*</font></h6></div>*

*<div class="col-9"><input type="text" name="email" class="form-control" required></div>*

*</div>*

*<br/>*

*<div class="row">*

*<div class="col-3"><h6>Message<font color="red">\*</font></h6></div>*

*<div class="col-9"><textarea name="msg" class="form-control" rows="4" required autocomplete="off"></textarea></div>*

*</div>*

*<br/>*

*<div class="row">*

*<div class="col-3"></div>*

*<div class="col-9">*

*<input type="submit" name="send" class="btn w-100" style="background-color:#D4AC0D;color: white; " value="SEND">*

*</div>*

*</div>*

*</div>*

*<div class="col-3 container">*

*<h5 style="color: #D4AC0D;">CONTACT US</h5><br/>*

*<h6><font color="#D4AC0D">Phone</font></h6>*

*<p>+375 123456789</p>*

*<h6><font color="#D4AC0D">Email</font></h6>*

*<p>email\_12@gmail.com</p>*

*<div style="color:#D4AC0D">*

*<i class="fab fa-facebook fa-2x"></i>&nbsp;<i class="fab fa-twitter-square fa-2x"></i>&nbsp;<i class="fab fa-instagram fa-2x"></i>&nbsp;<i class="fab fa-vk fa-2x"></i>*

*</div>*

*</div>*

*</div>*

*</div>*

*</form>*

*<?php include('footer.php') ?>*

*</body>*

*</html>Software Codes Add Hous*ing.

**A.7 Edit**

*<footer class="page-footer font-small stylish-color-dark pt-4" style="margin-top: 400px;background-color: #E5E7E9;">*

*<div class="container text-center text-md-left">*

*<div class="row">*

*<div class="col-md-4">*

*<h5 class="text-uppercase mb-4 mt-3 font-weight-bold">Content</h5>*

*<p>Student Housing Search System</p>*

*</div>*

*<hr class="clearfix w-100 d-md-none">*

*<div class="col-md-2 mx-auto">*

*<h5 class="text-uppercase mb-4 mt-3 font-weight-bold">Links</h5>*

*<ul class="list-unstyled">*

*<li>*

*<a href="index.php">Search</a>*

*</li>*

*<li>*

*<a href="contact.php">Contact</a>*

*</li>*

*<li>*

*<a href="about.php">About</a>*

*</li>*

*</ul>*

*</div>*

*<hr class="clearfix w-100 d-md-none">*

*<div class="col-md-2 mx-auto">*

*<h5 class="text-uppercase mb-4 mt-3 font-weight-bold">Links</h5>*

*<ul class="list-unstyled">*

*<li>*

*<a href="#!">Privacy</a>*

*</li>*

*<li>*

*<a href="#!">Help</a>*

*</li>*

*<li>*

*<a href="#!">FAQ</a>*

*</li>*

*</ul>*

*</div>*

*</div>*

*</div>*

*<?php*

*if (isset($\_SESSION['user\_logged\_in']) && $\_SESSION['user\_logged\_in'] = true) { }else{ ?>*

*<hr>*

*<div class="text-center py-3">*

*<ul class="list-unstyled list-inline mb-0">*

*<li class="list-inline-item">*

*<h5 class="mb-1">Register for free</h5>*

*</li>*

*<li class="list-inline-item">*

*<a href="signup.php" class="btn btn-danger btn-rounded">Sign up!</a>*

*</li>*

*</ul>*

*</div>*

*<hr>*

*<?php } ?>*

*<div class="text-center">*

*<ul class="list-unstyled list-inline">*

*<li class="list-inline-item">*

*<a class="btn-floating btn-sm btn-fb mx-1">*

*<i class="fab fa-facebook fa-2x"></i>*

*</a>*

*</li>*

*<li class="list-inline-item">*

*<a class="btn-floating btn-sm btn-tw mx-1">*

*<i class="fab fa-twitter fa-2x"></i>*

*</a>*

*</li>*

*<li class="list-inline-item">*

*<a class="btn-floating btn-sm btn-tw mx-1">*

*<i class="fab fa-instagram fa-2x"></i>*

*</a>*

*</li>*

*<li class="list-inline-item">*

*<a class="btn-floating btn-sm btn-tw mx-1">*

*<i class="fab fa-vk fa-2x"></i>*

*</a>*

*</li>*

*</ul>*

*</div>*

*<div class="footer-copyright py-3 text-center">*

*© 2018 Copyright: DANIEL DIEPREYE DAN*

*</div>*

*</footer>*