TALLINN UNIVERSITY OF TECHNOLOGY
Information technology department

TTU IT KOLLEDŽ

Kanyuchka Iakov 175797IDSR

TOUR BOOKING SYSTEM ANALYSIS AND DEVELOPMENT

PAKETTREISIDE BRONEERIMISE SÜSTEEMI ANALÜÜS JA ARENDUS

Bachelor thesis

Supervisor Nadežda Furs-Nižnikova
MBA

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud bakalaurisetöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Iakov Kanyuchka

16.05.2020

Annotatsioon

Bakalaureusetöö skoopis on analüüsida turismisektori, täpsemalt reisi broneerimise veebirendusi ja analüüsida ning arendada broneerimisi võimaldav veebirakendus. Eesmärk on luua reisi broneerimise veebirakendus, mis on muuhulgas ka hästi optimeeritud otsingumootorile (SEO). Lõputöö eesmärk on täies mahuses täidetud.

Antud bakalaureusetöö tulemusena valmis töötav veebirakendus, millele järgnes autori poolt koostatud analüüs ning tööriistade valik arendustöödeks.

Lõputöö on Inglise keeles ja sisaldab 37 lehti tekstiga, 4 peatüki, 52 joonist, 3 tabelit.

Annotation

The bachelor work thesis is focused on analyzing tourism sector, tour booking applications and implementing and developing own booking application. The purpose is to develop tour booking application well optimized for search engines. Application is ready to use.

The result of current bachelor thesis is a developed application, done with correctly chosen tools after well analyzation.

The thesis is in English and contains 37 pages of text, 4 chapters, 52 figures, 3 tables.

Table of Contents

Autorideklaratsioon..2

Annotatsioon..3

Annotation..4

Pictures list..7

Tables list..9

Annotation..11

Introduction..12

1 Research of the tourism sphere..14

1.1 The essence and basic concepts of tourism..14

1.2 Brief information about the travel company..15

1.3 Analysis of the organizational structure of the travel agency..16

2 Analysis of travel agencies websites..18

2.1 Analysis of travel agencies websites..18

2.2 Modern technologies of web-site development..19

2.3 Information system environment development..27

2.4 Localhost development tools..28

2.5 Data base development tools..29

3 Analysis of booking applications..31

3.1 Modern tour booking systems..31

3.1.1 Characteristics of reservation systems in tourism..31

3.2 Methods of reservation systems development..32

4 Tour booking web-application design and development..34

4.1 Information system data base design and development..34

4.2 Information system structure and funtctionality algorithms..34

4.3 Information system moduls creation..35

4.3.1 Registration module in system..35

4.3.2 Authorization module in system..35

4.3.3 “Tours” module 35

4.3.4 “Cart”module 35

4.3.5 “Order” module 36

4.4 Information system testing 36

Summary 37

Kokkuvõtte 39

Used literature list. 40

Feature 1: Picture list 41

Feature 2: SQL-script 67

Pictures list

Picture 1 The organizational structure of a travel agency........16
Picture 2 Upack result........41
Picture 3 Main settings........41
Picture 4 Automatical changing in hosts file blocking........41
Picture 5 Modules menu........42
Picture 6 tour model table........42
Picture 7 category model table........42
Picture 8 user model table........43
Picture 9 role model table........43
Picture 10 cart model table........43
Picture 11 order model table........43
Picture 12 Data base model........44
Picture 13 Information system data base structure........44
Picture 14 tour table structure........44
Picture 15 user table structure........45
Picture 16 role table structure........45
Picture 17 order table structure........45
Picture 18 category table structure........45
Picture 19 cart table structure........45
Picture 20 Data base model on the MySQL server........46
Picture 21 Web-application structure........47
Picture 22 Registration to system algorithm........48
Picture 23 Authorization algorithm in system........49
Picture 24 Tour adding algorithm........50
Picture 25 Registration form code........51
Picture 26 Registration to system code........52
Picture 27Authorization form code........53
Picture 28 Authorization form processor’s code........53
Picture 29 Tour list display table code........54
Picture 30 Tour adding form code........55
Picture 31 Tour adding form processor’s code........56
Picture 32 Tour editing form’s code........56
Picture 33 Tours editing form processor’s code........57

Picture 34 Tour category list’s display table code....................57
Picture 35 Category adding form’s code....................58
Picture 36 Category adding form’s processor code....................58
Picture 37 Category editing form’s code 59
Picture 38 Category editing form’s processor code....................59
Picture 39 Data added to cart by user code 60
Picture 40 File with tour added to cart updated information....................60
Picture 41 Code for file with the confimed order from cart....................61
Picture 42 File adding the orde to data base code 61
Picture 43 Registration page....................62
Picture 44 Authorization page....................62
Picture 45 Tour list page 63
Picture 46 Tour adding page....................63
Picture 47 Page updated....................64
Picture 48 Category list page....................64
Picture 49 Category adding page....................65
Picture 50 Category update page 65
Picture 51 User cart....................66
Picture 52 User orders....................66

Tables list

Table 1 The most prevalented CMS 25

Table 2 The most prevalented programming languages on web servers 25

Table 3 The most prevalented web servers 25

Annotation

The work is devoted to the study of the tourism sector, web applications in this area and the development of its own application.

International legislative organizations and scientific points of view are given. During work was made segmentation of types of tourism. Four basic functions of tourism are highlighted. Developed booking web application company-stakeholder organizational structure.

Was made analysis of sites and applications to these sites of leading tourism organizations.

Analysized main tools for developing own booking web application.

Developed own web application for booking tours.

Introduction

Tourism, as a leisure activity, has existed for many centuries. Since ancient times, simple human curiosity has attracted people to travel without any specific purpose. Today tourism is actively developing, people began to travel meaningfully - for educational, religious, recreational purposes. Cultural tourism has become an effective means of mass demand and wide availability of cultural and recreational values. Improvement the standards of living of the population and expanding the possibilities of free movement around the world have contributed to the emergence of millions of new travelers. Tourism began to play such a role in society that it turned into a unique large-scale global industry, and tourism resources are becoming an important part of the national wealth of many countries.

Being one of the main components of quality recreation, tourism is one of the elements of quality living conditions and helps to restore human life potential. Today, travel and tourism have become one of the most significant industries in the world and, to facilitate the organization of their leisure, there are many tourist destination websites available on the Internet that help you choose and book travel services.

Tourism resources have been developed using various programming languages and technologies and can both provide only the display of certain data, and organize interactive interaction with the user, as well as save and display large amounts of data. All these resources simplify the activities of travel companies, increase the level of customer service, and provide increased profits. In this regard, the study of the organization of a tourist Internet resource is an urgent issue in the functioning of any tourism enterprise.

Object: travel agency.

Subject: web application for booking tourist tours.

Purpose of work: to develop a web application for booking tourist tours.

Tasks:

- research of the tourism sphere;
- analysis of travel agencies websites;
- analysis of booking applications;
- design an application for booking tours;
- development of an application for booking tours.

In the process of performing the work, the following research methods will be used: analysis of scientific and educational literature, system analysis, comparative analysis, descriptive method, generalization method, synthesis, modeling.

The structure of the work consists of: introduction, four chapters, summary, list of references, applications.

1 Research of the tourism sphere

1.1 The essence and basic concepts of tourism

Tourism must be considered as a multifaceted and complex phenomenon, which includes various social relations that arise between a significant number of entities. It should be noted that tourism is an object of study of most sciences of sociology, economics, jurisprudence and others.

Analyzing the definitions of tourism presented in the special literature, A.E. Tolstova states that at present, three most common interpretations of tourism can be distinguished. Firstly, narrowed, reducing tourism to the activities of only tourism companies themselves. Secondly - one-sided consumer - when tourism is considered as a process of consumption of goods and services in their free time from production concerns outside the boundaries of their permanent place of residence. Thirdly, tourism is a different relationship in society regarding the use of people's free time outside their place of residence, in conjunction with certain conditions for the reproduction of a wide range of goods and services that are necessary for this [15].[1]. To a large extent, this approach has an economic focus.

A. P. Bogatov subdivides all definitions of tourism into highly specialized and conceptual (essential) ones. The first group of definitions concerns certain aspects of tourism (economic, social, legal, etc.) and acts as a tool for solving specific problems (for example, the definition of tourism for statistical purposes). The definitions assigned to the second group reveal the internal content of tourism, expressed in the unity of the whole variety of properties and relations, and make it possible to distinguish it from similar, often interrelated, but foreign phenomena [3][2].

According to the international association of scientific experts in the field of tourism, tourism is a set of relations and phenomena that arise during the movement and stay of people in places other than their permanent place of residence and work. Most of the definitions of tourism that characterize this concept through travel, travel, the movement of people who at the same time possess certain signs should be attributed to highly specialized [16] [3].

This approach is mainly used in legal sources. During the existence of various international tourism organizations, the definition of tourism has repeatedly changed, improved.

[1] Tolstova A.E. Tuorism. – Sankt Petersburg: Neva, 2013. – 410 p.

[2] Bogatov A.P. Safety in tourism. - M.: Infra-M, 2010. – 298 p.

[3] Tkachev V.A. Features of the specification of tourist resources of developed countries // International relationships. – 2015. - N3. – p. 32-37

The Hague Inter-Parliamentary Declaration on World Tourism of 1989 defines tourism as a phenomenon that includes all the free movement of people, as well as a service sector created to meet the needs arising from these movements [7] [4].

Another definition of tourism, often used by experts, was formulated by the UN Statistical Commission in 1993 and is contained in the Recommendations on Tourism Statistics of the World Tourism Organization. In these sources, tourism is defined as the activity of people who travel and stay in places outside their usual environment for a period not exceeding one consecutive year, for the purpose of recreation, as well as for business and other purposes [4] [5].

Based on the above definition of tourism, we can distinguish several of its main features. Firstly, the temporality of movement. Secondly, departure from a permanent place of residence. Thirdly, the presence of certain tourist goals. Fourthly, the prohibition of a traveler to engage in activities paid from a source in the country (place) of temporary residence. This feature stems from the economic nature of tourism. Tourism should bring a very specific economic benefit to the locality or to a country with tourism resources that attract tourists.

Today, one of the main trends in the tourism business is the transition from group forms to individual travels, when consumers plan and organize their trips on their own, using mainly the possibilities of Internet resources.

1.2 Brief information about the travel company

The main goal of a travel company is to maximize profits, achieve long-term cooperation with customers who contacted the company, as well as an individual approach to each client.

The main objectives of a travel company are:

1. Take a niche in the competitive tourism market;
2. Develop an individual approach to each client;
3. To improve the informationonal and material base of the company;
4. Improvement of the qualification of the personal.

A travel agency was created to carry out entrepreneurial activity, that is, independent activity aimed at the systematic profit. The company specializes in tourism activities, within which the company carries out the function of a travel agent. A distinctive feature of the company is the

[4] Pisarevski E.L. TOurism. Features and problems. – M.: Infra, 2011. – 430 p.

[5] Geography of tourism / Edited by professor A.J. Aleksandrovna. - M.: Knorus, 2011. – 475 p.

creation of a new travel service that allows customers to book vouchers online through the site without leaving home, which distinguishes the company from other companies.

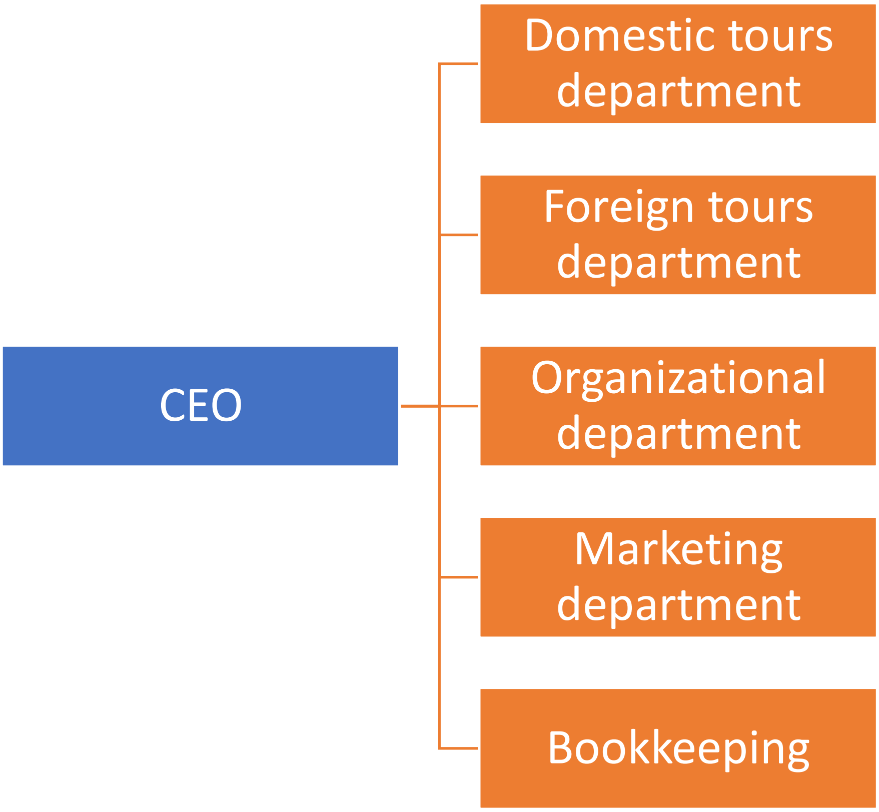
The travel agency acts as a tour agent for the sale of Estonian and foreign tours, therefore all clients of the travel agency are organized into two groups:

- tourists traveling on domestic routes;
- tourists traveling on external routes.

Each client group has a special approach and appropriate service - both group and individual. Tourists can also purchase individual tours, including for family holidays.

1.3 Analysis of the organizational structure of the travel agency

The organizational structure of the enterprise is the core that allows to optimally distribute responsibility and authority within the organization. The organizational structure of a travel agency is shown in Picture 1.



Picture 1 The organizational structure of a travel agency[6]

Tourism managers organize tours, make agreements with tour operators, send tourists (domestic tourism), and also buy and sell tours from tour operators (foreign tourism). Thus, the specialists

[6] Made by author

of these departments are engaged in the direct implementation of the tour operator and agent functions of the company.

The marketing department carries out the order of advertising on television, radio, print media, and Internet resources to think over a marketing strategy for the further development of the business.

The organizational department is responsible for organizing the activities of the travel agency, its logistical and information support, ensure the daily functioning of the office, and also make HR activities.

The accountant combines the functions of an accountant and an economist, performing all financial calculations, planning the economic performance of a travel agency, and is responsible for its financial statements.

CEO manages and controls all areas of the company. Heads of departments are directly accountable to him, and their authority includes solving tasks related to the work of departments, achieving each department’s planned targets, ensuring quality customer service, maintaining partnerships, etc. The director’s feedback with department heads is carried out as part of weekly meetings, at which emerging issues are considered, management decisions are made, plans are being adjusted.

2 Analysis of travel agencies websites

2.1 Analysis of travel agencies websites

Today the presence of a tourist enterprise site is considered not just a matter of prestige, but a necessity.

Novatours has its own website (https://www.novatours.ee/). Site page size: 16.1 kb (16436 bytes). Server response time: 0.8 seconds. The main page of the travel agency website has its own corporate identity. Here is the Novatours logo. The page opens with a multimedia presentation in the form of a slide show. Presentation is simultaneously the key to the site, with one click you can get information about the services provided. Below is a brief information about the travel agency. The text contains many links, which open next pages.

The site contains extensive information about the travel agency, its employees, tours and prices for tours, booking tours, hotels. The information is divided into sections: for agencies, legal entities. Each section provides relevant information. In the section for agencies, an agency agreement is presented, questionnaires and documents for consulates, advertising tours, financial support, catalogs, seminars and presentations, newsletter subscription [6][7].

Novatours is the largest tour operator in the Baltics. The main goal of the company is to offer passengers high-quality and reliable travel services and a wide selection of destinations. The main activity of Novatours is the organization of holiday and cultural trips, both charter and bus trips (Lithuania). Novatours Group offers almost 27 destinations a year. Charter flights start from Tallinn, Riga and Vilnius. In Estonia and Latvia, the tour operator operates under the Novatours brand, in Lithuania under the Novaturas brand.

Alfatour has its own website (https://alfatour.ee/). Site page size: 16.1 kb (16436 bytes). Server response time: 0.8 seconds. The main page of the travel agency website has its own corporate identity. Here is the Alfatour logo. The site has a bright color, contains a solid array of information necessary for the traveler: country descriptions, search for tours, hotel reservations, tickets and transfers. There are sections: water temperature at the main resorts of the world, information about shopping (articles, tips, news), etc. The site has contact information of the company; sections: "reviews", "main page", "price list"; proposals in various areas, a proposal for visas; for foreign guests, reception, registration, there is information for tourists in various countries.

[7] Karpova G.A. Tuorism resourses: problems and perspectives. – Sankt Peterburg: Neva, 2013. – 268 p.

2.2 Modern technologies of web-site development

A modern site is usually a dynamic site that allows the visitor to perform certain targeted actions on the site (submitting a form, registering on the site).

Web development uses many different technologies and products: these are programming languages and markups in a “pure” form, databases and other information storages, software products to simplify development (frameworks and site management systems), as well as server software. All of them are used to create sites. It should be noted that the most of modern sites are implemented using the web programming language php, but this option to create sites is very time-consuming and expensive.

All sites on the network can be of two types: static and dynamic sites. A page is called static that is entirely stored on the server and is displayed to the visitor in its unchanged form. A page is called dynamic that is formed by the server from several parts or by making (changing) data. All existing online stores are dynamic websites that may differ in development technology [1][8].

Consider the process of generating a dynamic page in more detail. If a static page does not undergo practically any changes and gets to the visitor in its “original” form, then a dynamic page, such as a person sees it, does not exist on the server. It is compiled in various ways from data stored on the server. With the advent of server-side programming tools, it became possible to make changes to the data sent to the visitor. This allows to change the content in response to the actions of the visitor. This can only be achieved by dividing the information and site design [8][9]. Using dynamic pages allows to store a certain design template, in which, depending on the page the visitor has visited, the necessary content is placed. This option is very convenient, because one or several templates are created for the entire site, and all design changes that need to be done on the site are made only with them.

Today there are many different technologies for creating dynamic sites, among which it is necessary to highlight the use of paid and free content management systems, frameworks, cloud platforms for creating sites, website programming using php and mysql, java and other technologies. Each of which has its own advantages and disadvantages, manifested in the efficiency of various services. In addition, when choosing a technology, the cost of its implementation is of great importance, which, if saving money necessary, can affect the further

[8] Aleksandrova, S. I. Website optimization technology as part of advertising campaign in the Internet URL: http://conf.sfu-kras.ru/sites/mn2011/thesis/s4/s4_120.pdf

[9] Kurzibova J.V. Dinamical websites development tools: training manual. – Irkutsk: IGU, 2011. – 121 p.

functionality of the site. At the same time, most developers when choosing a technology are guided by those known to them, and not by those that are optimal in a particular case. It is the technology with which the site was developed that affects its continued existence. It is important to take into account the prospects of a particular technology, and its relationship with various browsers and servers, as they important roles in the further effective operation of the website. Thus, the right choice of technology for creating a dynamic site opens up wider possibilities for using the site on the Internet, allows for timely updating of information on the site, and most importantly gives an undoubted advantage over competitors who have chosen unpromising technology [10][10].

Technologies for creating dynamic sites are generally divided into two groups: client-side technologies and server-side technologies. The main difference between these technologies is the place where the program code is processed. In the first case, all actions occur on the user's computer, and the effectiveness and correctness of operation depends on the selected browser and the characteristics of the computer. If certain problems occur, most users will not be able to fix them. In the second case, all requests from the client computer are transmitted to the server where the data is processed. As a result, a web page is formed, which is transferred back to the client's computer. When using this option, the processing capabilities are significantly expanded [13][11].

Client-side technologies:ActiveX controls and loadable modules; Java applets; client-side scripts and dynamic HTML.

Server-side technologies: ASP.NET; ASP; PHP; ColdFusion; JavaServer Pages и Java Servlets; Perl; Python.

An analysis of sites located on the Internet shows that the most used technologies for developing sites are PHP, ASP.NET, Perl and Python. It is worth noting that only ASP.NET technology is distributed on a paid basis. Only this technology requires a specialized server, as well as a special development environment. It is more difficult to master, which is justified by its focus on the development of large projects. However, the undoubted advantage is the power of the object approach and the support of the largest software developer [17][12].

[10] Maslyak T.A. Sites developing methods analisys.- Innovation technologies: science, techics, technology, education, health. – Kharkov, 2015 –230 p.

[11] Rublevskaya J.V. Business modeling in Internet., 2001. - 87-102 p.

[12] Shishkevich A.V. Modern technologies of website development. – Minslk: MMU, 2015. –110–114 p.

Other technologies are freely distributed, work on almost any server. The PHP technology, which is the most widely used among the freely distributed at the moment, is based on the PHP language, server expansion modules, and other components (databases, etc.). This allows the developer to choose the most suitable components, modify and freely distribute the code. In addition, a number of advantages can be distinguished, sites created using this technology: high capacity; availability of interfaces to many different database systems; built-in libraries for many common Web related tasks; low cost; relative simplicity of learning and using; cross-platform; source code availability.

When choosing a technology for creating a dynamic site, important to focus on issues related to the cost of introducing the technology. An important role is played by the performance of the technology when using various platforms. The prospect of technology development affects the continued availability of the site. It’s worthwhile to approach the choice of website development technology in a comprehensive manner, taking into account all the factors and the particularity of the resource being created [10][13].

A content management system is a software product that was created to simplify the creation of the main types of sites. Often control systems are quite universal, it gives the possibility to compare anything on them: from an information site to an online store or portal. At the same time, the CMS does not impose any restrictions on the appearance of the public part of the site (on what the ordinary user sees).

There are two types of CMS - boxed (finished software product that can be downloaded or purchased) and self-written (which can only be obtained from the developer) [6][14]. CMS can also be paid or free. Paid are developed by commercial companies and, often, are distributed in a package with certain conditions of support from the manufacturer. Most CMS are distributed under a license, which provides an unlimited period of use of the product and a limited period of update and support. Free CMS is developed by the community and distributed "as is", that is, they do not provide support from the manufacturer. Support can be obtained from other users of this system in various forums or in communities.

Frameworks are software products that simplify the creation and maintenance of technically complex or loaded projects. Most frameworks contain only basic software modules, and all

[13] Maslyak T.A. Sites developing methods analisys.- Innovation technologies: science, techics, technology, education, health. – Kharkov, 2015 –230 p.

[14] Karpova G.A. Tuorism resourses: problems and perspectives. – Sankt Peterburg: Neva, 2013. – 268 p.

project-specific components are implemented by the developer on their basis. This ensures not only high development speed, but also high productivity and reliability of solutions.

Popular php-frameworks: Yii, Symphony, Zend, Kohana, Codelgniter, CakePhp. Most popular ruby framework: Ruby on Rails.

Object environments (frameworks) represent the next level of improvement in web application development. Instead of combining markup and logic into a single module, object environments (frameworks) support the principle of separating content from presentation. The modules responsible for creating content are separated from the modules that display this content in a specific format [2][15].

The content management system is a software product, or rather, an automated information processing and management system that allows to develop both static and dynamic web sites / web portals and manage them by separating the processes of creating the contents of web sites / web portals and mechanisms of publish and format content on the Internet. Currently, CMS is a powerful functional automated system that allows to create not only blogs, but also complex multifunctional web-portals, which are widely used by users of various qualifications to solve their Internet problems [12][16].

The main tasks that modern CMS solve are the following:

- Automation of the entire development and publishing cycle of web applications on the Internet;
- Automated changing of the contents on the web portal pages on the network, providing the ability to implement various page layout designs by using different information display templates, plus the ability to widely use templates created by other developers;
- Differentiation of user authority, which allows independent configuration of the web portal for the presentation of different categories of users, including the ability to maintain personal pages;
- Ability to create a web portal inside a web portal, which allows to create complex web portals;
- Simplicity of editing and creating information for the web portal;
- The ability to automatically upgrade the web portal to new Internet standards;
- The ability to allow or prohibit the publication of information without removing it from the system;
- The ability to integrate and use project management tools for developing web applications;

[15] Amiragyan L.M. The choise of CMS systems for e-commerce / Amiragyan L.M, Pantikin D.S. Economics and sociality No 2(33) 2017 URL: http://publikacia.net/archive/2017/1/3/21

[16] Plusses and minuses. 5 popular internet-shops engines. URL: https://hostiq.ua/blog/5-popular-cms-for-e-commerce

- Scalability, which allows to develop systems by connecting software extensions from different developers;
- Multilingualism, the ability to automatically translate and support the web portal for different national target audiences;
- Diffirent website vercions support;
- Providing the opportunity to promote a web portal on the Internet by adjusting to search engine mechanisms such as Google, Yahoo, Yandex, etc.

The implementation of the above tasks leads to:

- improvement of the development quality;
- reducement of development cost and time;
- reducement of operating costs;
- reducement of the requirement for the support personel.

The main objective of the content management system is to create a platform for managing the structure and content of the website.

Typically, a CMS consists of two applications: CMA, a content management application, and CDA, a content delivery application. CMA allows a non-HTML author to control the creation, modification (including deletion) of website content without the help of a webmaster. To create a new page, the user does not need to have special knowledge: the text editor used in the CMS allows to organize site updates at the level of working with office applications. CDA module compiles generated content to update website.

CMA tools save the received document, CDA provide automatic layout of Web-pages based on predefined website design templates. Redesign can be done by simply replacing templates. CMS flexibility and intuitive interface make it easy to change the site structure and navigation system [5][17].

CMS has a modular structure. The most common modules are: news, guestbook, forum, catalog, chat. Thanks to this, a quick edition of the site to provide the necessary functionality structure is possible, which reduces the time and cost of modernization. For example, if it became necessary to post a large number of photos on a website, then simply connect the photo gallery module; if necessary, to conduct a survey of website visitors it is enough to connect the voting module. Specialized modules allow to optimize the publication of structured data, for example, price lists. CMS is a very convenient tool for creating information sites of a typical structure.

[17] Gornakov S.G. Mastering the popular site management systems (CMS). – M.: DMK Press, 2009. – 336 p.

The advantage of using CMS is also the reduction in the cost of creating and maintaining Internet sites, which is achieved by reducing the cost of the salary of the Web master, as well as time to search for documents that make up the content, stop duplication and eliminate errors.

When analyzing CMS, 9 main characteristics are distinguished:

1. System features: (CMS development language, database type (MySQL, MSSQL, Oracle, etc.), compatibility with web servers (Webservers: Apache, IIS)).

2. Safety: (SSL support (can this system be used with SSL certificate), confirmation of e-mail (whether confirmation keys are sent), management of visitors' sessions (does the system provide the administrator with information about visitors who are currently on the site)).

3. Support: (certification program, availability of user and administrator manuals, community, online help, third-party developers).

4. Simplicity of use: (drag-n-drop-content (can a user place content on a page in the style of drag-n-drop), friendly links (creating easily readable and accessible links to search engines), the ability to resize images, macro language (the presence of a language that allows the use of powerful software tools without deep skills), mass file uploads, saving the state preceding the changes).

5. Management: (advertising management, content management (the ability to post or delete) on a schedule, a trash (is it possible to recover information after deletion), themes / skins).

6. Compatiblity with content creation and sharing technologies: (RSS/XML, FTP, UTF-8, XHTML).

7. Flexibility: (support for cgi mode, extensible user profiles, interface localization, metadata, multilingual support).

8. Implementation: (advanced caching (navigation, templates), page caching).

9. Built-in applications: (blog, chat, forum, document management, calendar, guestbook, newsletter, photo gallery, voting, RSS (Really Simple Syndication).

There are currently dozens of commercial and free CMS in the world. In this regard, the problem arises of which CMS it is necessary to focus on in commercial activities in order to give users broad universal theoretical and practical for developing and deploying applications on the Internet [5][18].

Table 1 shows statistics on the use of CMS on the world Internet according to a study by Q-Success (Software Quality Management Consulting) on the World Wide Web Technology Surveys website.

[18] Gornakov S.G. Mastering the popular site management systems (CMS). – M.: DMK Press, 2009. – 336 p.

N/n	Name	Usage
1	WordPress	24,1%
2	Joomla	2,8%
3	Drupal	2,1%
4	Blogger	1,1%
5	Magento	1,1%

Table 1 The most prevalented CMS

As can be seen from table 1, the most prevalented CMS in the global Internet practice are WordPress, Joomla and Drupal.

World statistics on programming languages used on the server side of web applications are shown in Table 2.

N/n	Name	Usage
1	PHP	81,9%
2	ASP.NET	16,9%
3	Java	3%
4	ColdFusion	0,7%
5	Ruby	0,6%

Table 2 The most prevalented programming languages on web servers

Table 2 shows that the most popular programming language is PHP, which leaves far behind other programming languages for web applications.

From the point of view of the web servers used, the most prevalent is the Apache web server, as can be seen from table 3 below.

N/n	Name	Usage
1	Apache	57,1%
3	Nginx	24,6%
2	Microsoft-IIS	13,2%
4	LiteSpeed	2,2%
5	Google Servers	1,3%

Table 3 The most prevalented web servers

Based on the analysis of Table 3, we can conclude that the most prevalented database management system (DBMS) for deploying applications on the Internet is MySQL, since both WordPress and Joomla and Drupal are initially and currently focused mainly on the use of

MySQL. In the latest versions of these CMS, it became possible to use such DBMSs as PostgreSQL, MSSQL [11][19].

Thus, based on the above statistics, the choice of CMS can be reduced to a comparison of WordPress and Joomla.

The main language for the implementation and programming of WordPress and Joomla is widely used PHP, which allows to use both regular and object-oriented programming for web applications, and that successfully allows to use modern approaches in programming to develop web applications. Both systems use the well-established MySQL DBMS. At the same time, Joomla has a wider list of supported DBMSs than WordPress. DBMSs are relational and use the SQL programming language. The theory and practice of designing automated information processing and management systems used in the development of sites is well applicable in case of using Joomla and Wordpress.

An important factor is that WordPress and Joomla are open source and scalable. Plus they make it possible to use custom code to implement web applications. This allows users not only to use standard functionality, but the ability to develop their own code for the functioning of applications [20][20].

In general, the characteristics of both systems are quite similar, if not mention the line of DBMS and web-servers, as well as functionality, target audience, target area of application.

Plusses of developing a site using CMS:

- based on CMS, it’s quite simple and quick to create those types of sites that are embedded in the functionality of this system. Moreover, the development of simple solutions often does not even require the participation of a programmer in a project;
- many business processes that are needed for end users or are required to manage the site are already embedded in the CMS functionality. For example, within the framework of the predefined functionality, the site on the CMS already "knows" to edit pages and manage their structure, manage system users and their access level, sort and filter various objects, and much more;
- CMS website support is simpler than working with systems written from scratch, this is facilitated by both the availability of documentation and the high professional level of CMS developers, which is reflected in the logical and understandable system architecture;
- simple scaling is possible within the framework of the embedded functionality. For example, you can quickly add a product catalog to the information site, and then make an online store based on the catalog.

[19] iTrack official website. URL: http://itrack.ru/research/cmsrate

[20] WordPress official website. URL: http://wordpress.org

Minuses of developing a site using CMS:

- the implementation of non-standard functionality or fine-tuning for specific business processes, as a rule, is difficult. The processes embedded in the system are usually closely related, therefore, the modification of one of them leads to the need to modify those dependent on it. And in some cases, the implementation of specific functionality within the CMS is completely impossible;
- the flip side of CMS versatility is the redundancy of functionality and the accompanying increase in the complexity of management and use, as well as some performance problems (a site on a CMS performs guaranteed more data operations than the process requires, and the data storage structure is usually somewhat “bloated” relative to real project needs).

The considered technologies for creating sites allow to quickly create high-quality sites, but the created site will contain functionality that is not relevant for the company, increase the load on the server of the system, thereby increasing the requirements for hardware resources. In this regard, the development of our own content management system for a travel company is the most optimal option, as it will allow us to create a system designed exclusively for a certain area of activity.

2.3 Information system environment development

To develop a web application, PHPStorm is selected. This is an integrated development environment for PHP with an intelligent editor, supports PHP 7.0, 5.6, 5.5, 5.4 and 5.3 for projects, provides code completion, refactoring, error prevention and supports language mixing [93].

There are many reasons why PHPStorm was chosen as the development environment. Intuitive interface, wide functionality, backlight support, support for a wide range of databases. The program allows to access the database and enter information there, as well as make changes to the data in it. In addition, the program supports downloading and uploading project files from the server, as well as updating and editing files.

The program supports advanced web development technologies, including HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, Jade templates, Zen Coding, Emmet, and, of course, JavaScript. PHPStorm includes WebStorm features (HTML / CSS editor, JavaScript editor) and gives full-featured support for PHP and database / SQL.

Common features of the PHPStorm program [19][21]:

[21] PhpStorm – integrated development environment for PHP. URL: http://jetbrains.ru/products/phpstorm

- intelligent PHP code editor with syntax highlighting, code autofill, advanced code formatting settings, errors prevention;
- support of 7.0, 5.6, 5.5, 5.4 and 5.3, generators, coprograms and syntactical updates;
- PHP refactorings, code (re)arranger, duplicated Code Detector;
- support of Vagrant, Composer, built-in REST-client, Command Line Tools, SSH console;
- framewors support (MVC view for Symfony2, Yii) and specialized plagins for leading PHP frameworks (Symfony, Magento, Drupal, Yii, CakePHP);
- debugger for PHP applications, validation of debugger configuration, PHPUnit code coverage and integration with profiling programs;
- HTML, CSS, JavaScript editor. Debugging and unit testing for JS. HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, Emmet support;
- a complete set of tools for front-end development;
- code style support, integrated styles PSR1/PSR2, Symfony2, Zend, Drupal ect;
- application remote deployment and automated synchronization using FTP, SFTP, FTPS;
- Live Edit;
- PHP UML;
- Integration with bug tracking systems;
- Data base tools, SQL editor;
- cross-platform (Windows, Mac OS X, Linux).

2.4 Localhost development tools

To create an information system, the capabilities of a web server are required. The most convenient way to organize the functionality of the web server is to use tools that provide the capabilities of the web server on the computer on which the program is being developed.

One of the most most popular program packages to organize the local server is OpenServer. It is a portative server platform and program environment made for web developers [18][22].

The software package has a rich set of server software, a convenient, multi-functional thought-out interface, and has powerful capabilities for administering and configuring components. The platform is widely used for the development, debugging and testing of web projects, as well as for the provision of web services on local networks.

Possibilities of managing program: inconspicuous work in Windows tray, quick srat and stop, server starts automatically when program is launched, several domen managing modes, virtual disk mount, command promt managing support, profile settings support, usable components logs view, http, MySQL and PHP mõdule switch, detailed and understandable documentation, one-click access to domens, quick access to a configuration templites, multilanguage interface, programm autorun by list.

[22] Open Server official website. URL: http://open-server.ru

Complex features: don’t need installation, possibility to work from USB flash, synchronous work with Denwer, XAMPP, work in local/network/external IP address, SSL support without extra setups, domen is creating just by creation standart folder, server protection from external access, more than 40 portable programs package, task scheduler (cron), possibility to create a local subdomain without losing the visibility of the main domain on the Internet.

There are three versions of OpenServer: basic, premium and ultimate. Ultimate version unpacks not only webservers, but extra software (Picture 2).

“Domains” directory includes virtual hosts, “modules” foler – OpenServer modules (different versions of DBMS MySQL, interpreter of different versions of PHP and web-server itself (nginx или Apache). Inside üserdata folder modeles setups are stored.

After launchin OpenServer appears red flag icon at the tray (server stoped), green (server launched) or yellow (restart).

“Main” menu (Picture 3) shows main program package settings. It allows to create profile and to make different unique setups for each profile needed [18][23]. OpenServer automatically adds record to the hosts file. There is possibility to run server without editing errors (if there are any) in file (Picture 4).

OpenServer can provide not only local work, but can also make a personal computer a full-fledged web server.

To create a project, need to create a folder with the desired name in the domains folder, and create an index file in this folder - index.php, thus creating a virtual host in which the developed project will be located. If there are problems with the launch, or if necessary, find out the status of the web server, OpenServer provides the ability to view log, this allows to register, find and fix errors in the server. “Modules” menu is to server configurations (choose server, PHP version and DBMS (Picture 5)).

2.5 Data base development tools

For the web-application data base development the MySQLWorkbench inviroment was used. It is made for data bace visual design and MySQL server management. For the modules construction there is DataModeling section.

[23] Open Server official website. URL: http://open-server.ru

Program functionality: the program allows to visualize the database model in graphical form; visual, functional mechanism for establishing connections between tables, including many-to-many connections with the creation of a connection table; restoration of the table structures from an existing database server; user friendly SQL request editor, it allows to send request to the server and receive response in table view; possibility to edit data in table in visual mode.

For the data base development was used phpMyAdmin web-application too.

PhpMyAdmin – open source application, written on PHP code. It is a web-interface for DBMS MySQL administration. Allows to administrate MySQL servers, run SQL orders and wach table and data bace content from the web-browser [9][24].

PhpMyAdmin allows to: create and delete data bases; create, copy, delete, rename and edit tables; support tables; delete, edit and add margins; send SQL-request; manage keys; download text file to the tables; create graphical schemes of data bases in rbb format; search in data base and its sections.

[24] Brett McLaughlin. PHP & MySQL: The Missing Manual. – O'Relly Media, Inc, 2011. – 544 p.

3 Analysis of booking applications

3.1 Modern tour booking systems

Currently, there are several hundreds of tourist services reservation systems that can be divided according to their functional purpose: airline ticket reservation; hotel reservation; reservation of several tourist services in one system (airline tickets, hotels, car rental, etc.); booking tourist products created by tour operators.

Moreover, reservation systems can be classified according to users: reservation systems intended for professional market participants (travel agents); reservation systems available to end users (amateur tourists).

3.1.1 Characteristics of reservation systems in tourism

Anywayanyday reservation system (anywayanyday.com) is one of the most popular systems for tourists and is considered one of the best online services for booking travel and business trips. It has been operating since 2008 and provides the acquisition of airline tickets and hotel reservations worldwide. The system uses the resource of the global distribution system Amadeus. The system is available 800 airlines and 330 thousand hotels. On average, it takes 3 minutes to search, select and purchase an air ticket, and for each request, many flight options are offered, the system is updated in real time. In addition to the service for individuals, the system has special software solutions for web portals, corporate clients, agencies, in particular, Anywayanyday service for the sale of airline tickets can be installed on the website of any company. There is a mobile application system option.

Booking.com is one of the most famous reservation systems in the world, with more than 700 thousand nights booked daily on the site. Booking.com B.V. founded in 1996, its head office is located in Amsterdam (Netherlands), in 59 countries around the world there are representative offices. The system provides an interface in 40 languages. The reservation system contains more than 543,884 accommodation facilities in 207 countries. Accommodation objects are presented in different segments: hotels, apartments, campsites, villas, country houses, guest houses, botels, apart-hotels, chalets, hostels, capsule hotels, etc.

Agoda (www.agoda.com) - The system was created in 2005 and is owned and operated by Agoda Company Pte. Ltd registered in Singapore. It is currently one of the fastest growing online hotel reservation platforms in the world. The system contains 545 854 hotels around the world, implements an interface in 38 languages.

HRS.com is an international online portal that offers hotel reservations in 190 countries for individual tourists and corporate clients. The company was established in 1972 in Cologne, offers hotel reservation for private trips and a full range of hotel services for corporate clients: analytics and selection of hotel databases, optimization of travel costs, online booking of hotels and conference rooms, group accommodation, convenient intuitive interface, a detailed description of each hotel and the best price.

Hotels.com - more than 240 thousand hotels around the world are represented in the reservation system. Hotels.com LP is a subsidiary of Expedia. Hotels.com LP is headquartered in the USA (Dallas, Texas).

Airbnb.com - the service began operation in 2008 in San Francisco (California), provides search and booking of unique housing in 34 thousand cities and 190 countries.

iWay Express - transfer booking services in 400 cities of the world.

3.2 Methods of reservation systems development

In addition to the considered options for creating sites using content management systems, there are cloud systems that provide ample opportunities for organizing commercial electronic resources, while there is no need for any work with hardware, as it is provided by cloud service operators. Consider cloud systems that allow to implement the functionality of a booking system based on cloud technologies.

Jimdo is a free SaaS service with which anyone can create their own website. More than 12 million sites have already been created on the Jimdo service, on which users have created more than 200 thousand sites.

Jimdo offers a wide range of templates - ready-made and colorful solutions for a site of any subject. A free version of Jimdo will be available under the name: www.username.jimdo.com. Paid versions of Jimdo include additional templates, a domain name and Email addresses - all that is needed for a successful company presentation of on the network. The JimdoBusiness version has everything for creating an online store.

Platform WIX.COM. An international cloud platform for creating and managing Internet projects. The system is completely free and is suitable for creating an information site, as well as for a full online store.

The company also launched the App Market, where is possible to purchase various applications to expand the functionality of the site. Applications are both paid and free, developed by Wix and third-party companies. The App Market allows to integrate features such as photo feeds, blogs, playlists, online communities, emails and file managers, and much more [14][25].

The platform also has an extended paid business version of Premium, which allows user to connect his own domain to the site, remove Wix banners, get additional storage space, advertising coupons, etc.

Templates for creating a site are divided into categories: business and services, music, entertainment, online shopping, restaurants and hotels, photography, etc. Wix has a built-in function to create a mobile version of the site. But this function does not apply to all templates suitable for working on the mobile version; they are marked “compatible with mobile”.

The considered cloud technologies, although allow to quickly create electronic resources for trade, including for the sale of tourist tours, have a number of drawbacks that reduce the convenience and efficiency of their use, as well as the CMS have excessive functionality, in some cases necessary functions are available at paid rates, for their development it is necessary to conduct staff training, which can be inconvenient when changing the program. In this regard, the development of its own sales management system for a travel company is the best option for organizing an electronic sales platform for a travel company.

[25] Rublevskaya J.V. Business modeling in Internet., 2001. - 87-102 p.

4 Tour booking web-application design and development

4.1 Information system data base design and development

As noted capter 2.1, the capabilities of the MySQLWorkbench program were used to design the database. In Pictures 6-11 the model tables of the information system database are show.

Picture 12 shows data base logic model.

After data base model creation SQL-script creation is needed. running a script on a MySQL server will create a physical database model.

SQL-script is shown on the Feature 2 page.

After SQL-request was sended on the My SQL server data base was created tables for the information system. Data base structure is shown on the Pictures 13-19.

4.2 Information system structure and funtctionality algorithms

To understand which blocks and fuctcional will be in application important to design application structure and working algorithms. Only after that is codding may start.

Site structure includes main page. From its user can go to the pages: Tour categories; Tours list; User cart (for authorized users only); User orders (for authorized users only).

Picture 21 shows future web-application structure.

User registration algorithm is very important part of a modern web-application. It helps to organize communication with the site visitor. Registration allows user to add to the information system own data and with-it application will segregate tihat user from others and give him special funkcional and possibilities after authorization.

Picture 22 shows user registration algorithm on the site.

After user registration another algorithm is needed which will respond for user authorization. Picture 23 is showing authorization process.

Tour and categories adding to the information system pages is one of the main functions which is alloing to place on the web-pages categories and tours. Picture 24 show tours adding to site schemes.

4.3 Information system moduls creation

4.3.1 Registration module in system

The registration module allows users to register on the site and after that, with this data enter the site. Coding process is shown on the Pictures 25-26.

4.3.2 Authorization module in system

Authorization module in system is in the login.php file, there are authorization form (Picture 27) and authorization form processor (Picture 28).

4.3.3 "Tours" module

"Tours" module have: tour.php; tour-add.php; tour-delete.php; tour-edit.php; tour-edit-remove.php; tourincategory.php falise.

Inside tour.php tours list is generating tour list (Picture 29)

File tour-add.php responsible for adding tours to the data base (Picture 30).

Tour adding form's processor is in the tour-add.php form. Code is shown on the Picture 31.

File tour-edit.php is responsible for tour data update, in it are: tour editing form (Picture 32) and "Categories" module

"Categories" module have: category.php, category-add.php, category-delete.php, category-edit.php and category-edit-remove.php files.

In the category.php file tour categuries list is generating (Picture 34).

File category-add.php is responsible for adding categories to the data base. Inside this file are category adding form (Picture 35) and data processor from adding form (Picture 36).

For category data editing is responsible file category-edit.php, it have category editing form (Picture 37) and category editing processor (Picture 38).

4.3.4 "Cart"module

"Cart" module have cart.php and addincart.php file. Inside cart.php is proceeding adding tours to cart, which were sended there by user.

File addincart.php is for making a data record with information about what tour user added to cart to the data base.

4.3.5 “Order” module

“Order” module includes order.php, addorder.php files. Inside order.php file generating requests with orders which user was ordered inside the cart (Picture 41).

addorder.php file (Picture 42) is for adding orders to the orders table inside the data base. Its also sends e-mail to to company’s manager with the information about the order.

4.4 Information system testing

After finishing the modeules development it is necessary to test their capability. Pictures 44 – 52 show how the webpages are view on the brouser.

Summary

During the bachelor work the main target was solved. Web-application for tour booking is developed. While making to the main targed was solved the following tasks:

- the tourist sphere was analized;
- tourist companie’s websites were analized;
- booking applications were analized;
- tour booking web application was designed;
- tour booking web application was developed.

Tourist sphere is an internation worldwide segment. It creates a whole separate market segment, which allows to create a huge number of jobs. It brings additional income to related areas, such as culture, restaurant business, trade.

Tourism brings profit to the state. Tourist - cannot receive income from the state in which he arrived.

Tourism can also be educational in nature and at the moment, it is the educational sphere of tourism that falls into the growth factor of the tourism sector. It is also necessary to add to the peculiarity of the tourism industry its dependence on information technology, as more and more often people want to go to the site and make their own trip.

There are hundreds of thousands of travel companies in the world. This is a fairly competitive segment. The best of them have long spanned the whole world. At the same time, there are much more local firms than international ones. In order to gain a niche in this market, you must bring something of your own. Unique content, your website optimized for the most frequently used by residents of the state that the company wants to come to, will allow us to count on a positive result from the work of a startup these days.

Booking applications for tours are a must-have for a self-respecting travel agency. Of course, the best of them belong to the largest world players, such as Booking.com for an example. But local players are also beginning to actively use their developments. Designing a template for a local-made booking application is a good idea for promoting a firm in the local market.

As a Content Managment System author uses WordPress – it is most popular CMS well known by most search engines. To implement the web application author used Relational Data Base MySQL, because WordPress is initially and currently focused mainly on the use of MySQL. For visual design of the data base author used MySQL Workbench environment, because it allows to visualize the database model in graphical form, to immediately send requests to the server and receive a response in the form of a table. Ability to edit data in a table in visual mode makes it wery comfortable to use. For developing working interface for the data base was used PHP language possibilities it is a main language for WorldPress.

Kokkuvõtte

Põhisiht bakalaureusetöö ajal oli täidetud. Reisi plaanerimis veebirakendus on arenenud. Need on põhisihtmäärkid mis olid täidetud põhitöö sooritamisajas:

- Turismieriala oli analüseeritud;
- Turismifirmade kodulehed olid analüseeritud;
- Broneerisimis veebirakendus olid analüseeritud;
- Reisi bronerimis veebirakendus oli disainitud;
- Reisi bronerimis veebirakendus oli arendatud;

Turismi eriala on ülemaailmane äri segment, millel toimetavad paljud töötajad. See eriala annab suur lisa sissetulek seotatud valdkonnale nagu kultur, restoraniäri, kaubandus.

Turismiala toob tulu ka riigile. Reisi ei saa sissetulek riigist kuhu tema jõudis.

Turism võib oma olemuselt olla ka hariduslikku laadi. Turismitööstuse eripärale tuleb lisada ka sõltuvus infotehnoloogiast. Rohkem inimesi tahavad kohapeal plaanerida seiklused.

Maailmas on sajad tuhanded reisifirmasid. Reisieriala on üsna konkurentsitihe. Parimad nendest on juba ammu oma äri kasvanud ülemaailma tasemele. Samal ajal on kordades rohkem väikesed reisi ettevõtted, mis toimetavad ühel riigil. Selleks et oma koht sellel turul saada, ettevõtte peab oma uus idee levitama. Unikaalne sisuga ja hea optimiseeritud veebirakendusega sihtides kasutajad mis on pärit riigist kus meie hoiame focus, annab meile võimalus kohe arvestama positivne tulemus tööst kohe alustamis päevast.

Reisi broneerimis veebirakendus on kohustuslik edukale reisi ettevõttele. Muidugi kõige paremad nendest kuuluvad ülemaailma taseme ettevõttele nagu näiteks Booking.com. Kui aga ka lokalsed firmad hakkavad kasutama erinevad veebirakendus oma töös. Lokaalse äri veebirakenduse mustandi disainimine on hea idee selleks et edendama firma oma lokaalsel turul.

Autor avatud lähtekoodiga ajaveebi ehk blogi haldamise süsteem puhul kasutab WordPress – see on kõige levinum CMS. Veebirakenduse andmebaasi arendamise jaoks oli kasutatud Relatsiooni Andmebaas MySQL. Selle valiku põhjus on seda et WordPress kasutab MySQL. Arendamise mugavuse jaoks oli kasutatud MySQL grafiline interface MySQL Workbench keskkond, selle pärast et see töörist visualiseerib andmebaasi mudel grafilisel kujul. See töörist saadab otseselt käsud andmebaasi serverile ja saab ka kohe vastus tabeli kujul. Võimalus grafilises kasutaja keskonnas muuta andmebaas võimaldab väga mugavalt kasutada MySQL. Selleks et teha liides andmebaasi jaoks oli kasutatud PHP programeerimis keel, kuna PHP on WordPressi põhi keel.

Used literature list.

1. Aleksandrova, S. I. Website optimization technology as part of advertising campaign in the Internet URL: http://conf.sfu-kras.ru/sites/mn2011/thesis/s4/s4_120.pdf

2. Amiragyan L.M. The choise of CMS systems for e-commerce / Amiragyan L.M, Pantikin D.S. Economics and sociality No 2(33) 2017 URL: http://publikacia.net/archive/2017/1/3/21

3. Bogatov A.P. Safety in tourism. - M.: Infra-M, 2010. – 298 p.

4. Vasilchevsky A.S. Content managment system technology URL: http://belisa.org.by/pdf/PTS2005/219-220.pdf

5. Gornakov S.G. Mastering the popular site management systems (CMS). – M.: DMK Press, 2009. – 336 p.

6. Karpova G.A. Tuorism resourses: problems and perspectives. – Sankt Peterburg: Neva, 2013. – 268 p.

7. Kuznecov M.V. PHP by examples. – Sankt PEterburg: BHV, 2012. – 400 p.

8. Kurzibova J.V. Dinamical websites development tools: training manual. – Irkutsk: IGU, 2011. – 121 p.

9. Brett McLaughlin. PHP & MySQL: The Missing Manual. – O’Relly Media, Inc, 2011. – 544 p.

10. Maslyak T.A. Sites developing methods analisys.- Innovation technologies: science, techics, technology, education, health. – Kharkov, 2015 –230 p.

11. iTrack official website. URL: http://itrack.ru/research/cmsrate

12. Plusses and minuses. 5 popular internet-shops engines. URL: https://hostiq.ua/blog/5-popular-cms-for-e-commerce

13. Aleks Ross. HTML, CSS, PHP, MySQL. Traning manual, 2011 - 68 p.

14. Rublevskaya J.V. Business modeling in Internet., 2001. - 87-102 p.

15. Tolstova A.E. Tourism. – Sankt Peterburg: Neva, 2013. – 410 p.

16. Tkachev V.A. Features of the specification of tourist resources of developed countries // International relationships. – 2015. - N3. – p. 32-37

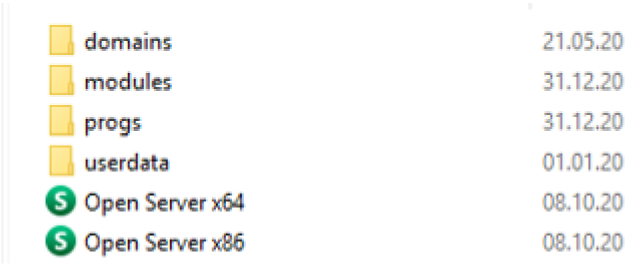
17. Shishkevich A.V. Modern technologies of website development. – Minslk: MMU, 2015. – 110–114 p.

18. Open Server official website. URL: http://open-server.ru

19. PhpStorm – integrated development environment for PHP. URL: http://jetbrains.ru/products/phpstorm

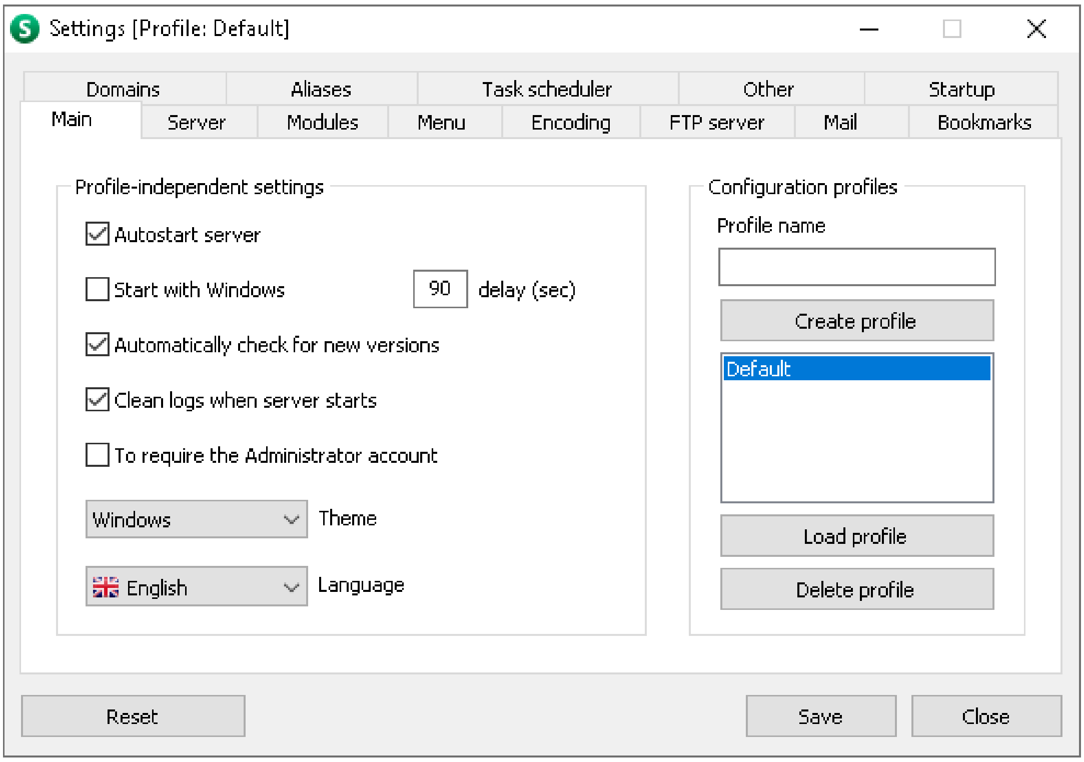
20. WordPress official website. URL: http://wordpress.org

Feature 1: Picture list



domains	21.05.20
modules	31.12.20
progs	31.12.20
userdata	01.01.20
Open Server x64	08.10.20
Open Server x86	08.10.20

Picture 2 Upack result[26]



Settings [Profile: Default]

Profile-independent settings

☑ Autostart server

☐ Start with Windows 90 delay (sec)

☑ Automatically check for new versions

☑ Clean logs when server starts

☐ To require the Administrator account

Windows Theme

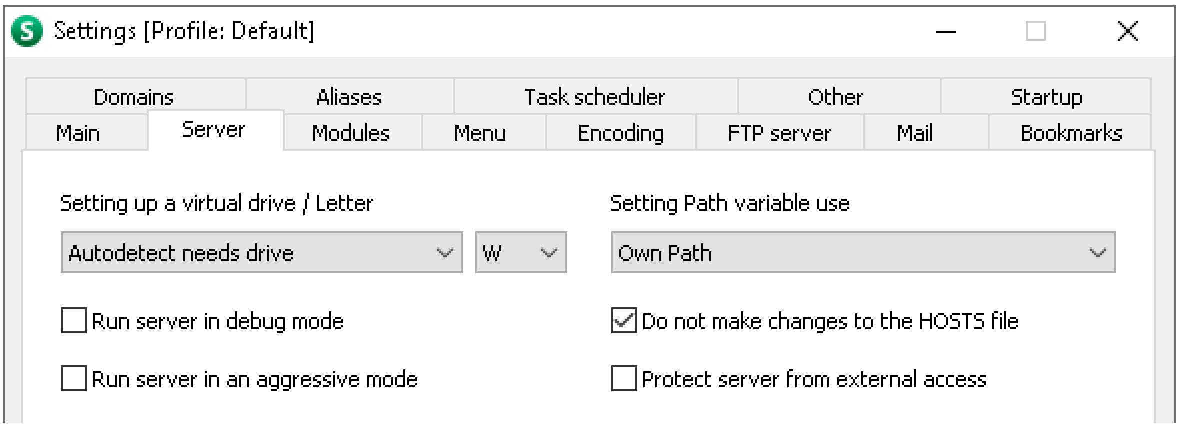
English Language

Configuration profiles

Profile name

Default

Picture 3 Main settings[27]



Settings [Profile: Default]

Setting up a virtual drive / Letter

Autodetect needs drive W

Setting Path variable use

Own Path

☐ Run server in debug mode

☐ Run server in an aggressive mode

☑ Do not make changes to the HOSTS file

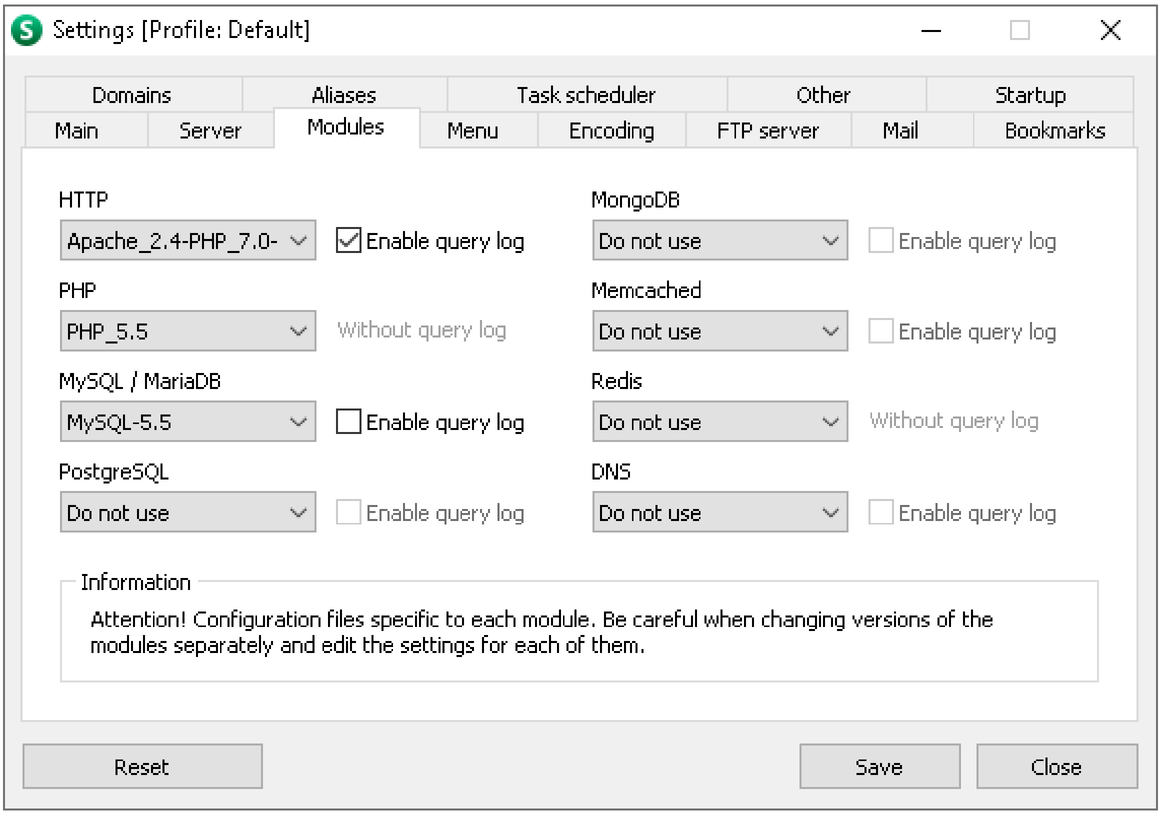
☐ Protect server from external access

Picture 4 Automatical changing in hosts file blocking[28]

[26] Author’s desktop skreen

[27] Author’s desktop skreen

[28] Author’s desktop skreen



HTTP: Apache_2.4-PHP_7.0- ☑ Enable query log

PHP: PHP_5.5 Without query log

MySQL / MariaDB: MySQL-5.5 □ Enable query log

PostgreSQL: Do not use □ Enable query log

MongoDB: Do not use □ Enable query log

Memcached: Do not use □ Enable query log

Redis: Do not use Without query log

DNS: Do not use □ Enable query log

Information

Attention! Configuration files specific to each module. Be careful when changing versions of the modules separately and edit the settings for each of them.

Picture 5 Modules menu[29]

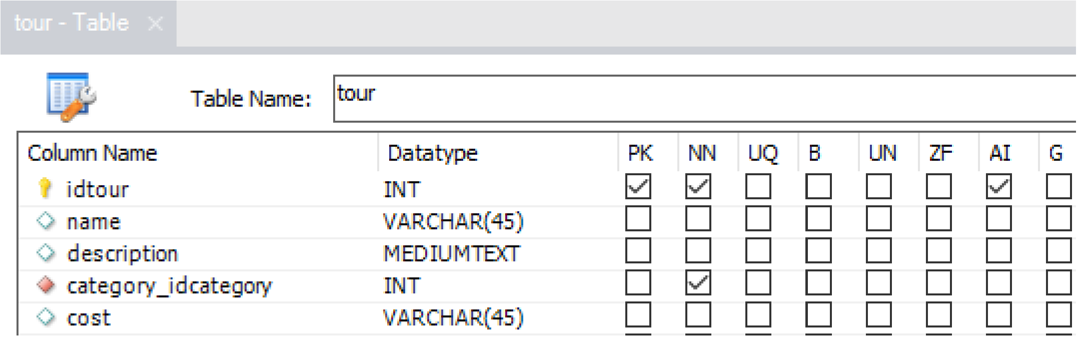


Table Name: tour

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idtour	INT	☑	☑	□	□	□	□	☑	□
name	VARCHAR(45)	□	□	□	□	□	□	□	□
description	MEDIUMTEXT	□	□	□	□	□	□	□	□
category_idcategory	INT	□	☑	□	□	□	□	□	□
cost	VARCHAR(45)	□	□	□	□	□	□	□	□

Picture 6 tour model table[30]

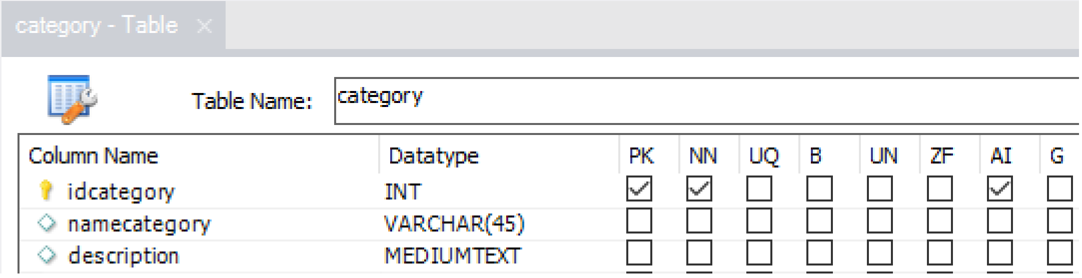


Table Name: category

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idcategory	INT	☑	☑	□	□	□	□	☑	□
namecategory	VARCHAR(45)	□	□	□	□	□	□	□	□
description	MEDIUMTEXT	□	□	□	□	□	□	□	□

Picture 7 category model table[31]

[29] Author’s desktop skreen

[30] Author’s desktop skreen

[31] Author’s desktop skreen

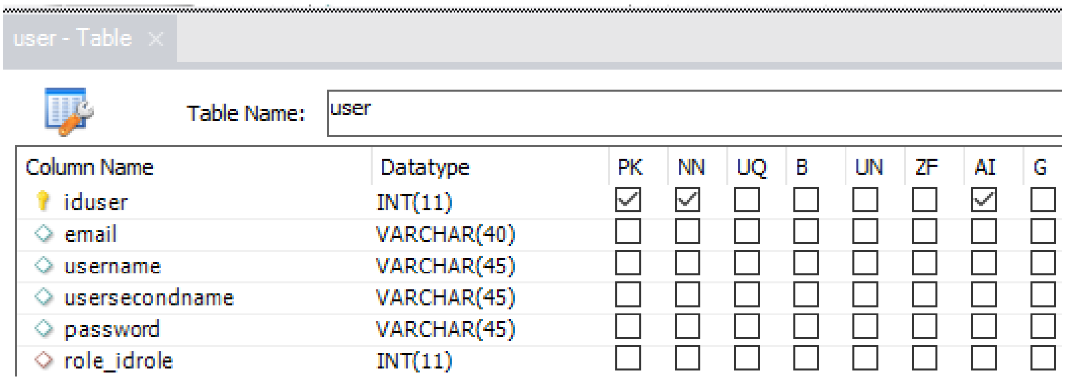


Table Name: user

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
iduser	INT(11)	☑	☑	□	□	□	□	☑	□
email	VARCHAR(40)	□	□	□	□	□	□	□	□
username	VARCHAR(45)	□	□	□	□	□	□	□	□
usersecondname	VARCHAR(45)	□	□	□	□	□	□	□	□
password	VARCHAR(45)	□	□	□	□	□	□	□	□
role_idrole	INT(11)	□	□	□	□	□	□	□	□

Picture 8 user model table[32]

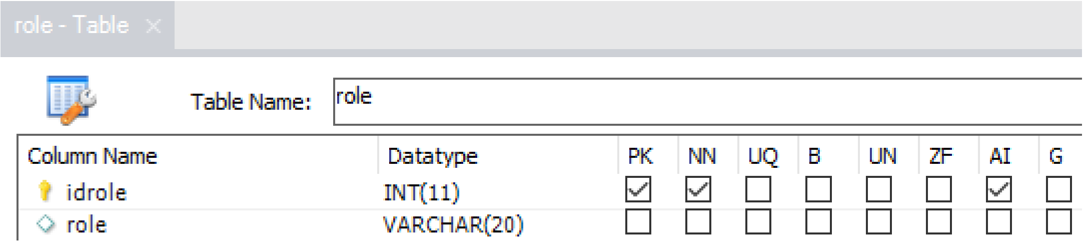


Table Name: role

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idrole	INT(11)	☑	☑	□	□	□	□	☑	□
role	VARCHAR(20)	□	□	□	□	□	□	□	□

Picture 9 role model table[33]

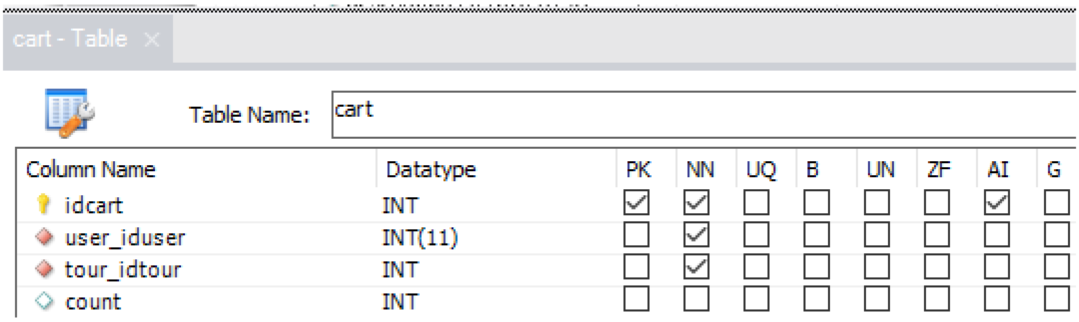


Table Name: cart

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idcart	INT	☑	☑	□	□	□	□	☑	□
user_iduser	INT(11)	□	☑	□	□	□	□	□	□
tour_idtour	INT	□	☑	□	□	□	□	□	□
count	INT	□	□	□	□	□	□	□	□

Picture 10 cart model table[34]

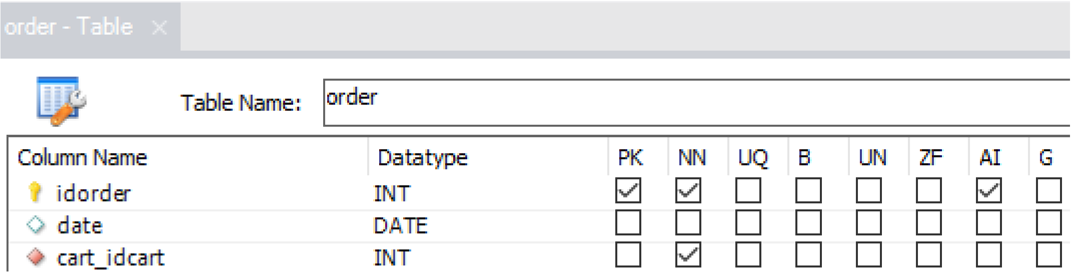


Table Name: order

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idorder	INT	☑	☑	□	□	□	□	☑	□
date	DATE	□	□	□	□	□	□	□	□
cart_idcart	INT	□	☑	□	□	□	□	□	□

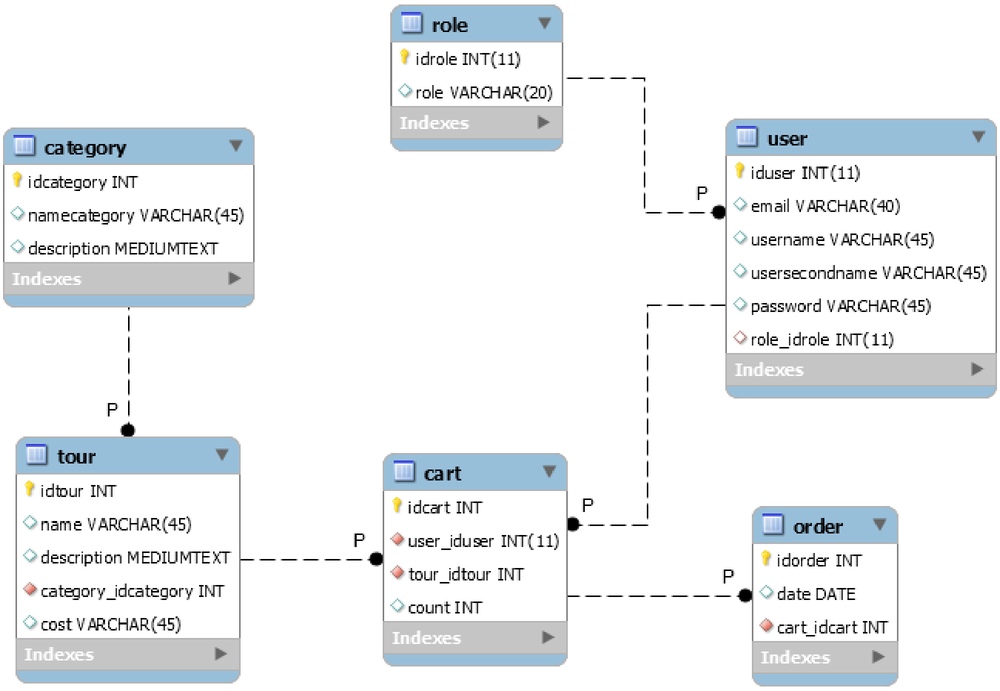
Picture 11 order model table[35]

[32] Author’s desktop skreen

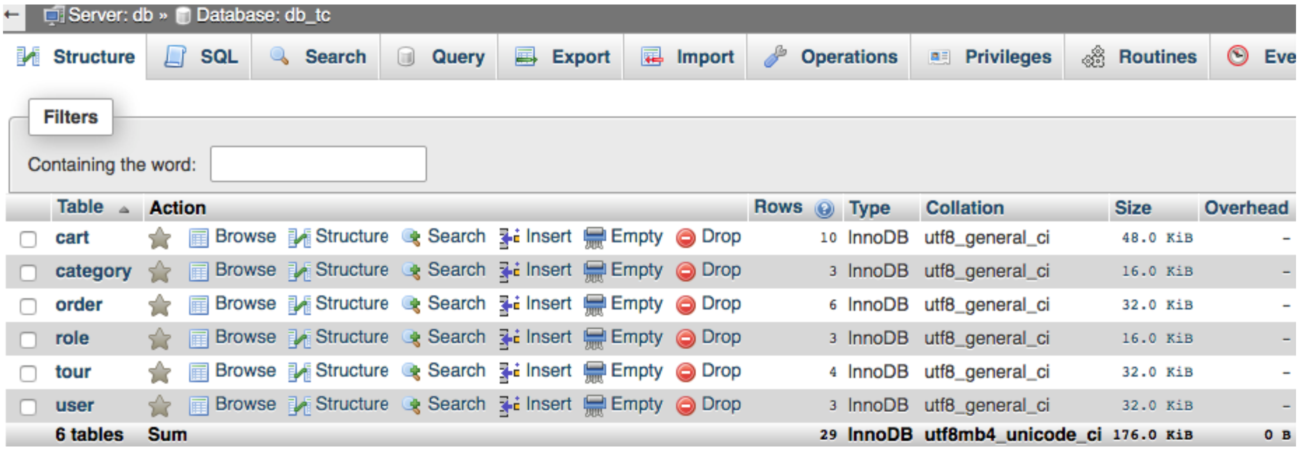
[33] Author’s desktop skreen

[34] Author’s desktop skreen

[35] Author’s desktop skreen



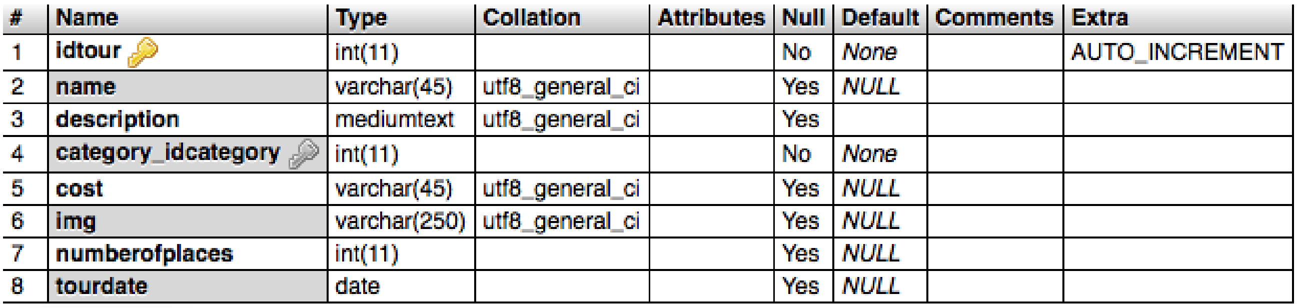
Picture 12 Data base model[36]



Containing the word:

	Table	Action	Rows	Type	Collation	Size	Overhead
□	cart	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8_general_ci	48.0 KiB	-
□	category	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16.0 KiB	-
□	order	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_general_ci	32.0 KiB	-
□	role	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16.0 KiB	-
□	tour	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	32.0 KiB	-
□	user	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	32.0 KiB	-
	6 tables	Sum	29	InnoDB	utf8mb4_unicode_ci	176.0 KiB	0 B

Picture 13 Information system data base structure[37]



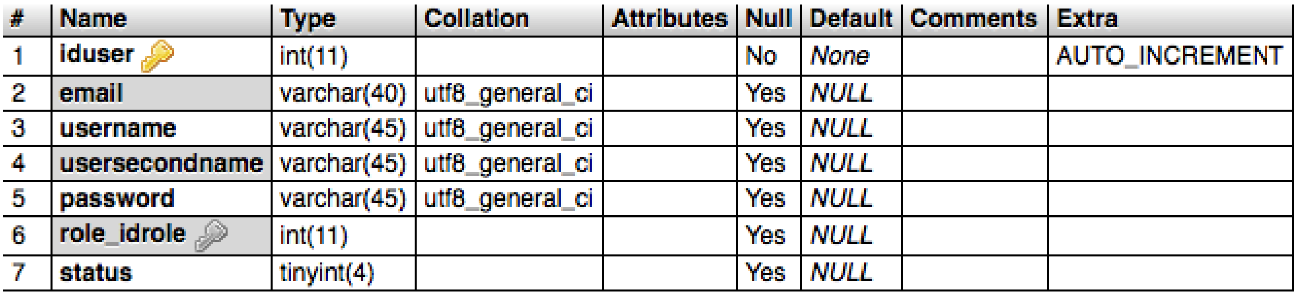
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	idtour	int(11)			No	*None*		AUTO_INCREMENT
2	name	varchar(45)	utf8_general_ci		Yes	*NULL*		
3	description	mediumtext	utf8_general_ci		Yes			
4	category_idcategory	int(11)			No	*None*		
5	cost	varchar(45)	utf8_general_ci		Yes	*NULL*		
6	img	varchar(250)	utf8_general_ci		Yes	*NULL*		
7	numberofplaces	int(11)			Yes	*NULL*		
8	tourdate	date			Yes	*NULL*		

Picture 14 tour table structure[38]

[36] Author's desktop skreen

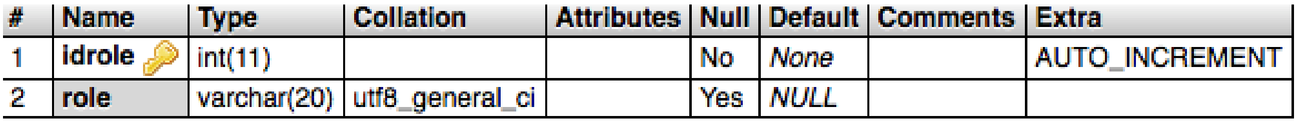
[37] Author's desktop skreen

[38] Author's desktop skreen



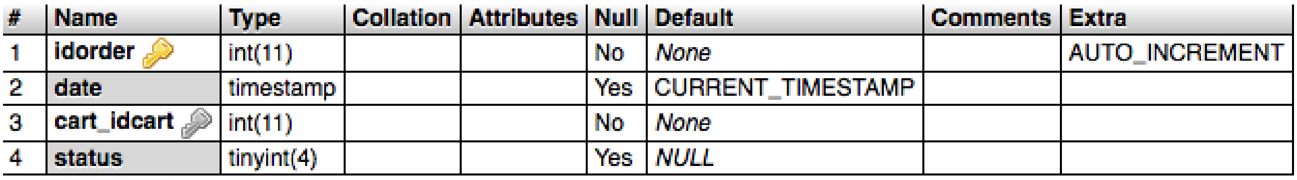
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	iduser 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	email	varchar(40)	utf8_general_ci		Yes	*NULL*		
3	username	varchar(45)	utf8_general_ci		Yes	*NULL*		
4	usersecondname	varchar(45)	utf8_general_ci		Yes	*NULL*		
5	password	varchar(45)	utf8_general_ci		Yes	*NULL*		
6	role_idrole 🔑	int(11)			Yes	*NULL*		
7	status	tinyint(4)			Yes	*NULL*		

Picture 15 user table structure[39]



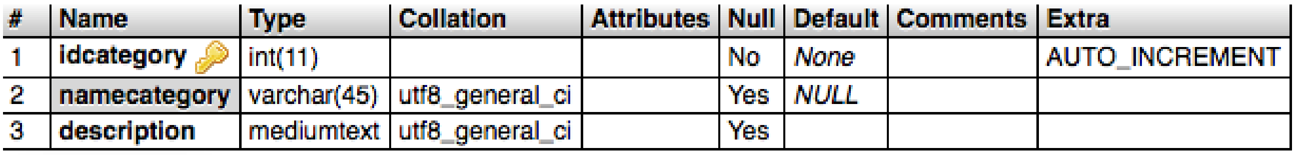
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	idrole 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	role	varchar(20)	utf8_general_ci		Yes	*NULL*		

Picture 16 role table structure[40]



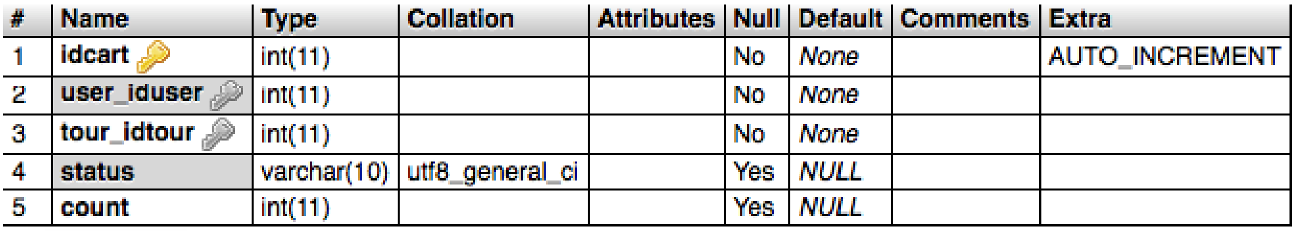
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	idorder 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	date	timestamp			Yes	CURRENT_TIMESTAMP		
3	cart_idcart 🔑	int(11)			No	*None*		
4	status	tinyint(4)			Yes	*NULL*		

Picture 17 order table structure[41]



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	idcategory 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	namecategory	varchar(45)	utf8_general_ci		Yes	*NULL*		
3	description	mediumtext	utf8_general_ci		Yes			

Picture 18 category table structure[42]



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	idcart 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	user_iduser 🔑	int(11)			No	*None*		
3	tour_idtour 🔑	int(11)			No	*None*		
4	status	varchar(10)	utf8_general_ci		Yes	*NULL*		
5	count	int(11)			Yes	*NULL*		

Picture 19 cart table structure[43]

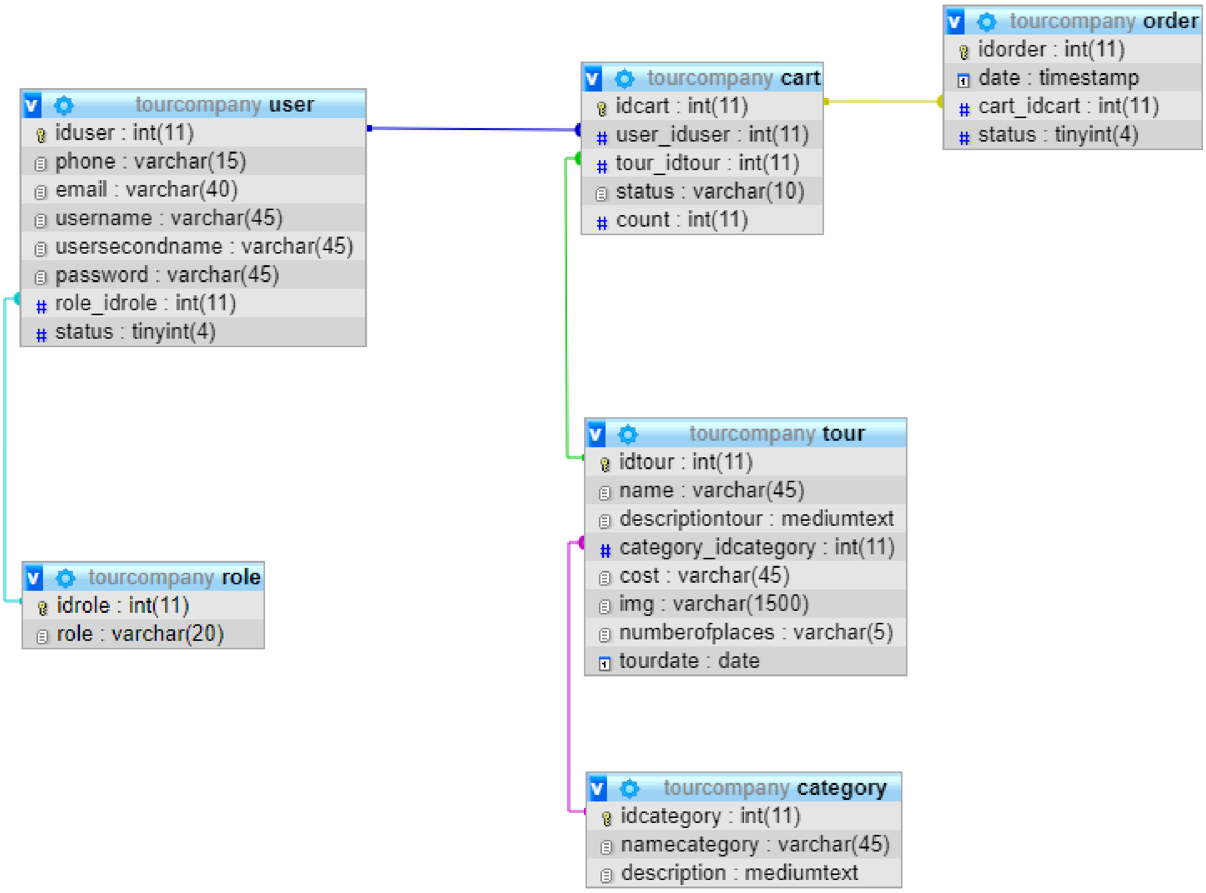
[39] Author's desktop skreen

[40] Author's desktop skreen

[41] Author's desktop skreen

[42] Author's desktop skreen

[43] Author's desktop skreen



tourcompany user
- iduser : int(11)
- phone : varchar(15)
- email : varchar(40)
- username : varchar(45)
- usersecondname : varchar(45)
- password : varchar(45)
- role_idrole : int(11)
- status : tinyint(4)

tourcompany cart
- idcart : int(11)
- user_iduser : int(11)
- tour_idtour : int(11)
- status : varchar(10)
- count : int(11)

tourcompany order
- idorder : int(11)
- date : timestamp
- cart_idcart : int(11)
- status : tinyint(4)

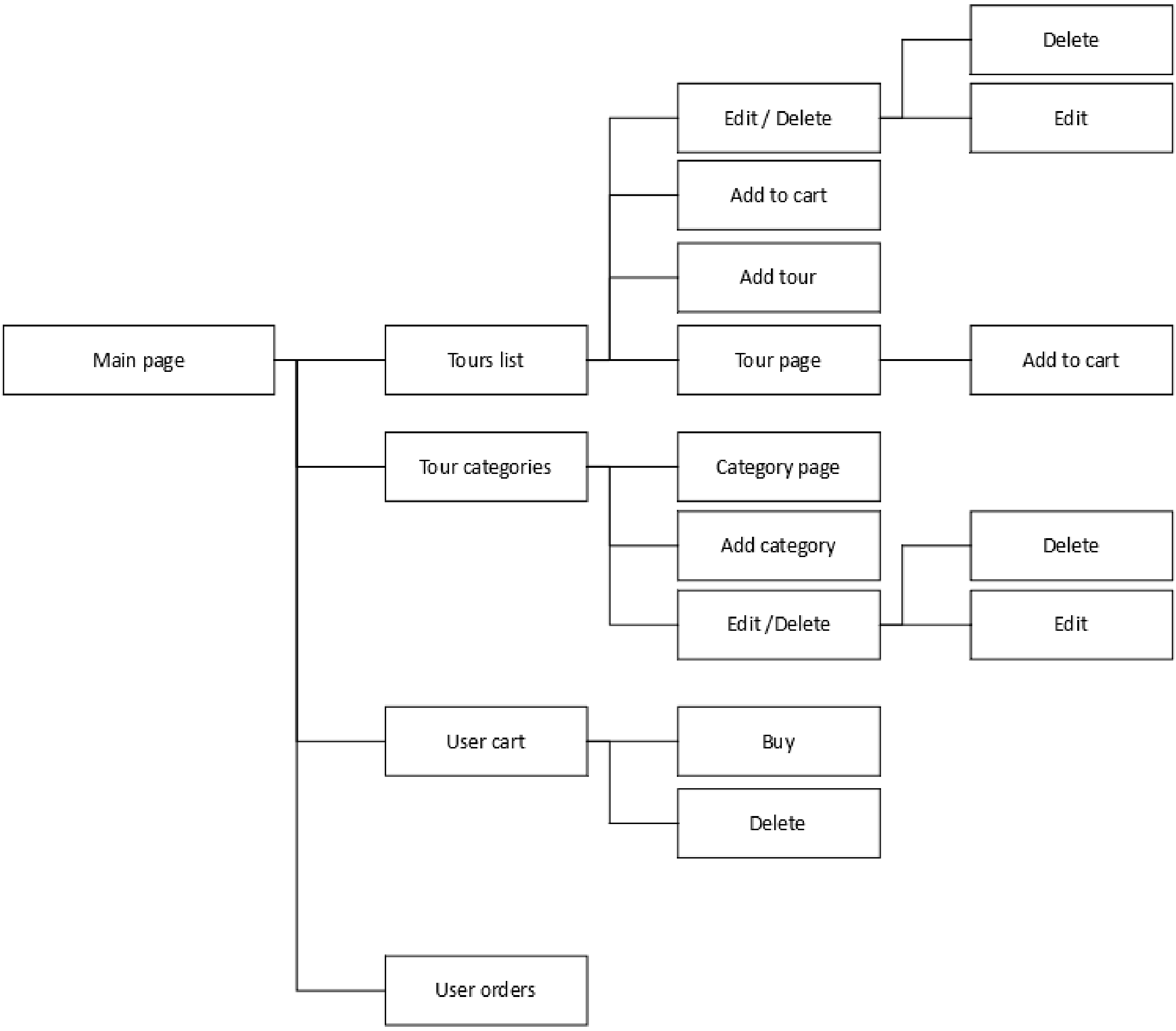
tourcompany role
- idrole : int(11)
- role : varchar(20)

tourcompany tour
- idtour : int(11)
- name : varchar(45)
- descriptiontour : mediumtext
- category_idcategory : int(11)
- cost : varchar(45)
- img : varchar(1500)
- numberofplaces : varchar(5)
- tourdate : date

tourcompany category
- idcategory : int(11)
- namecategory : varchar(45)
- description : mediumtext

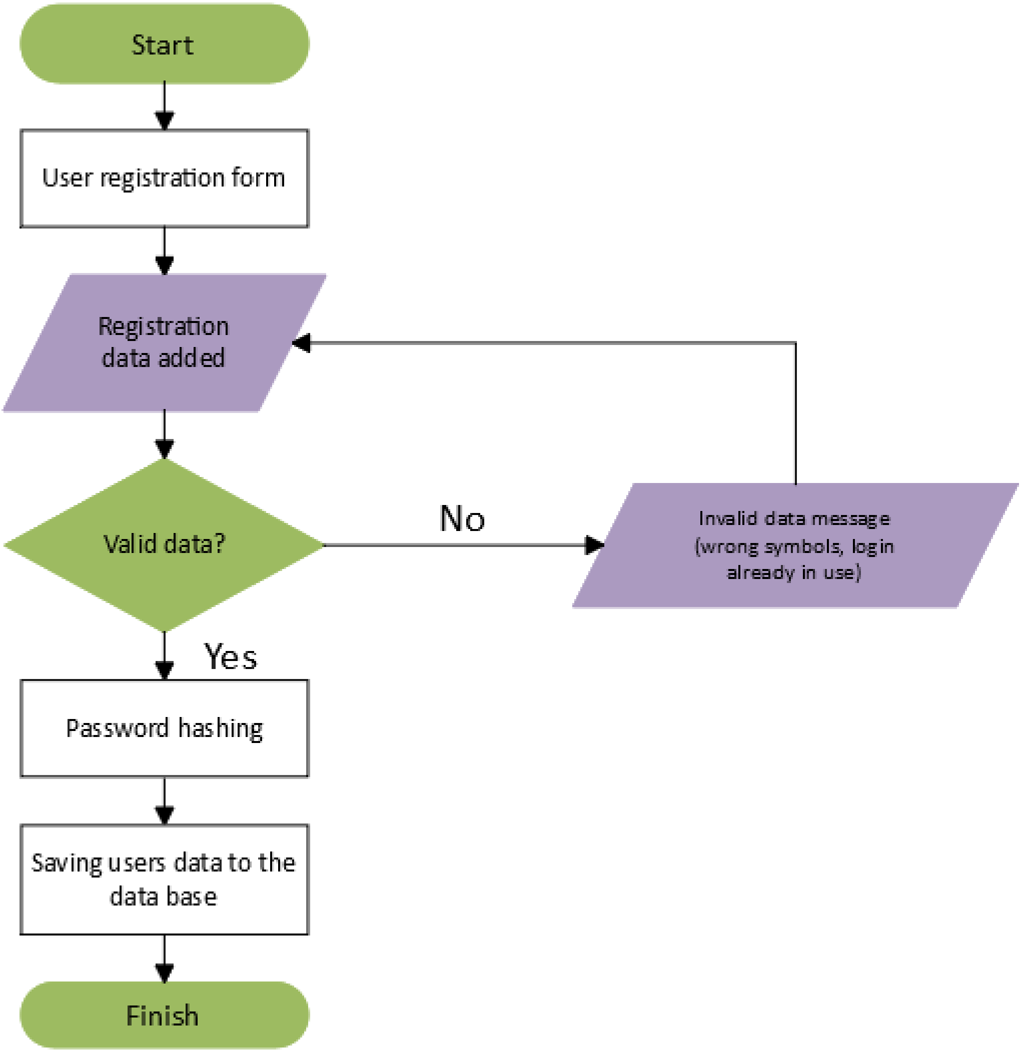
Picture 20 Data base model on the MySQL server[44]

[44] Author’s desktop skreen



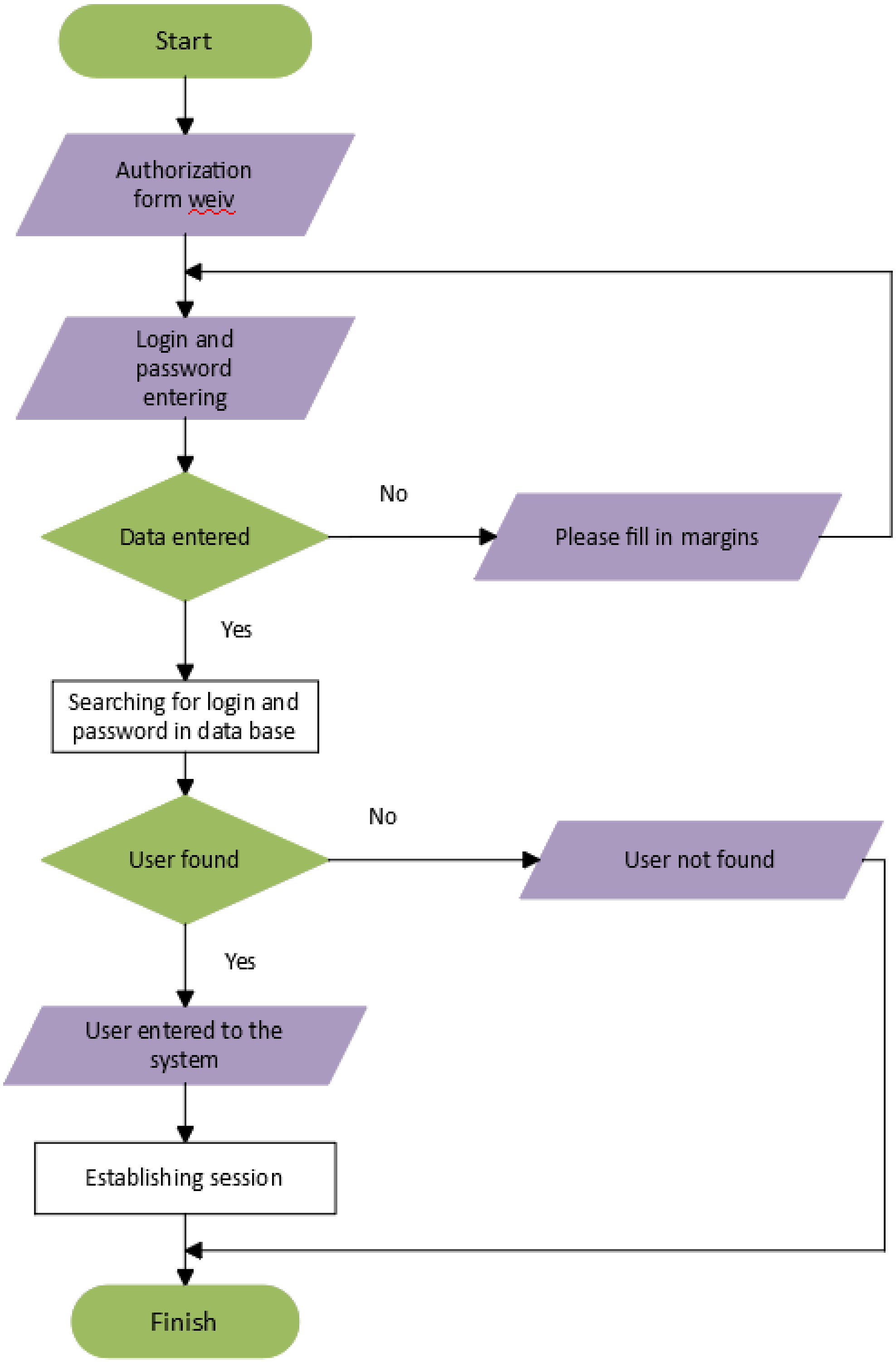
Picture 21 Web-application structure[45]

[45] Author’s desktop skreen


Start
User registration form
Registration data added
Valid data?
No
Invalid data message (wrong symbols, login already in use)
Yes
Password hashing
Saving users data to the data base
Finish

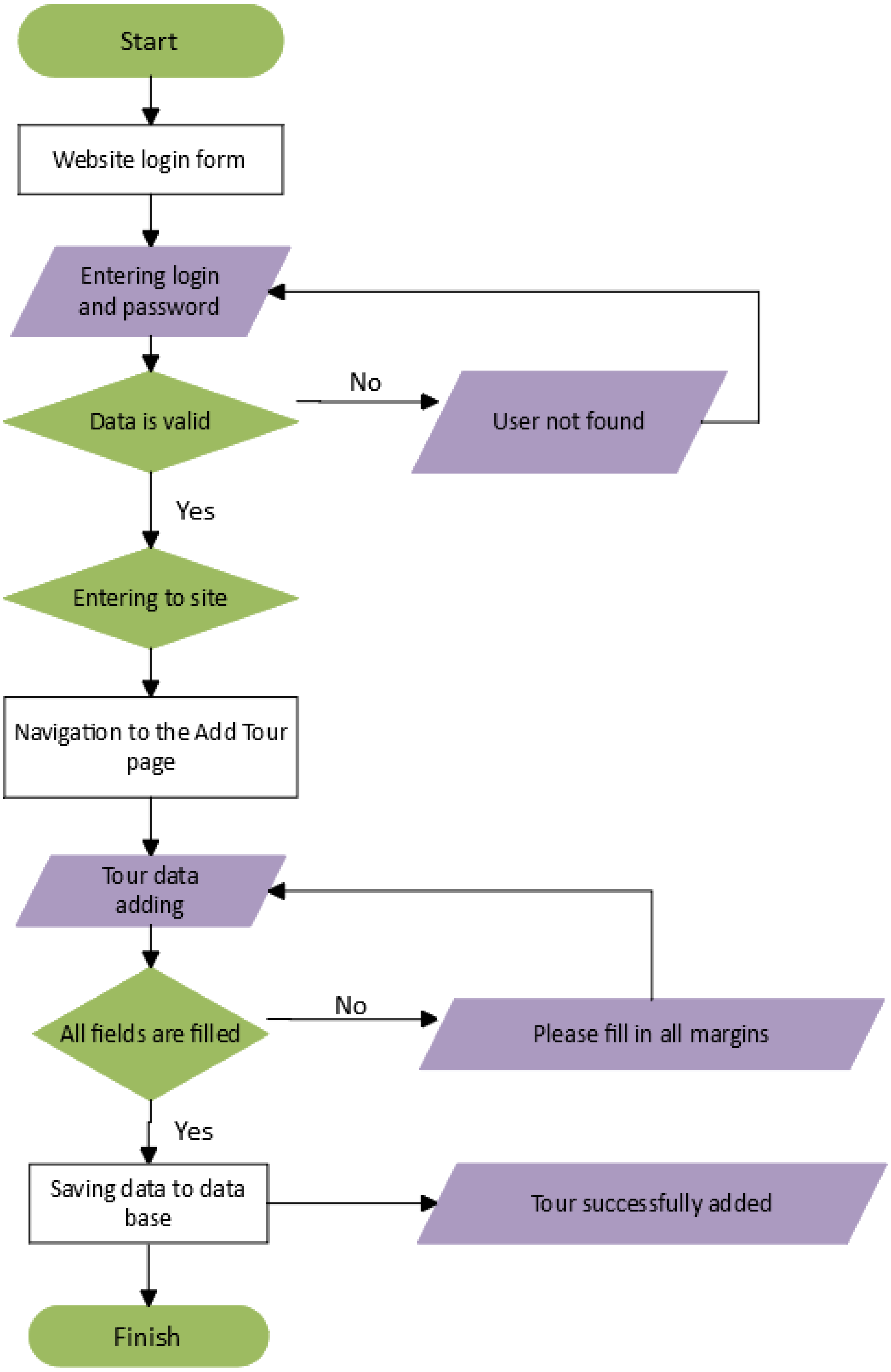
Picture 22 Registration to system algorithm[46]

[46] Author's desktop skreen



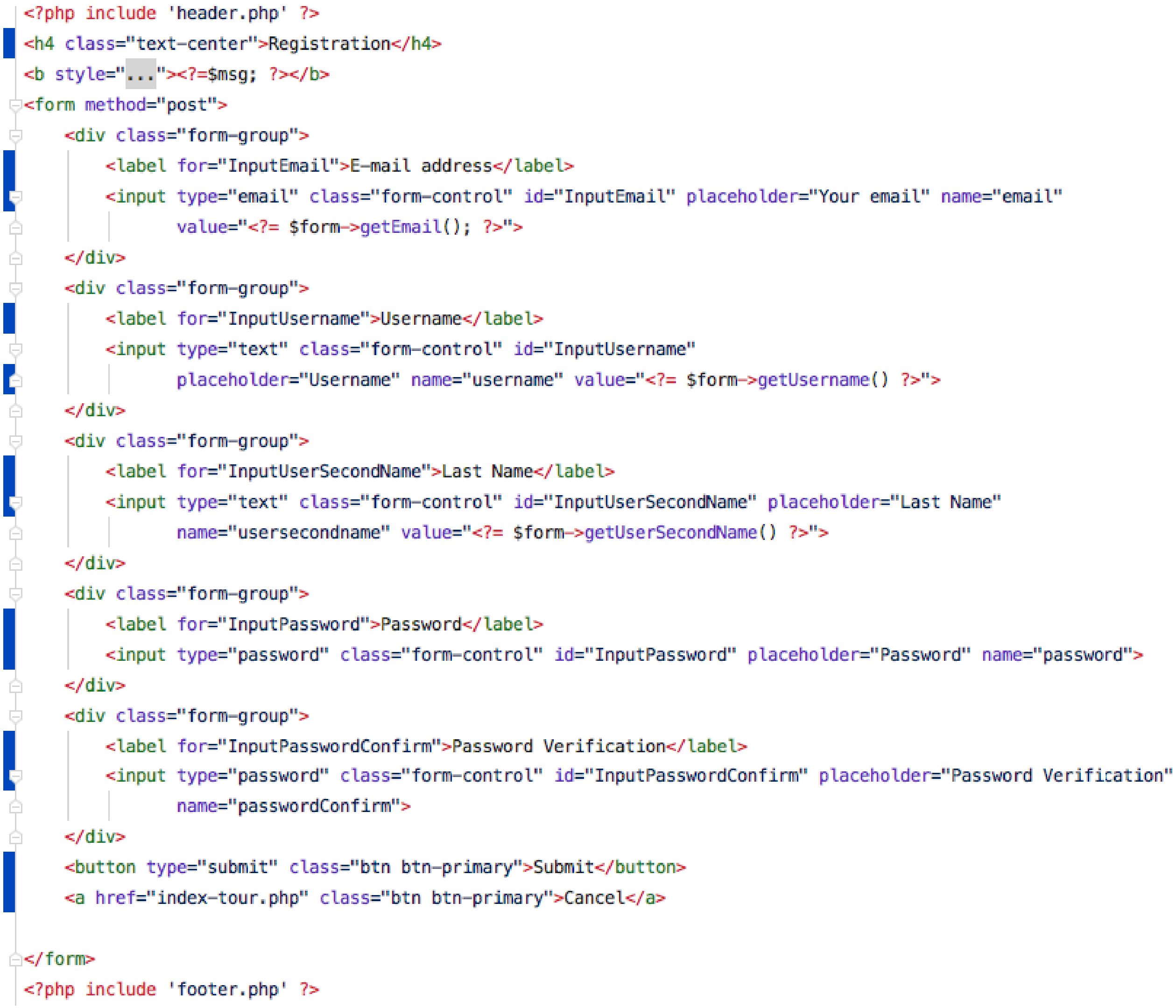
Picture 23 Authorization algorithm in system[47]

[47] Author's desktop skreen



Picture 24 Tour adding algorithm[48]

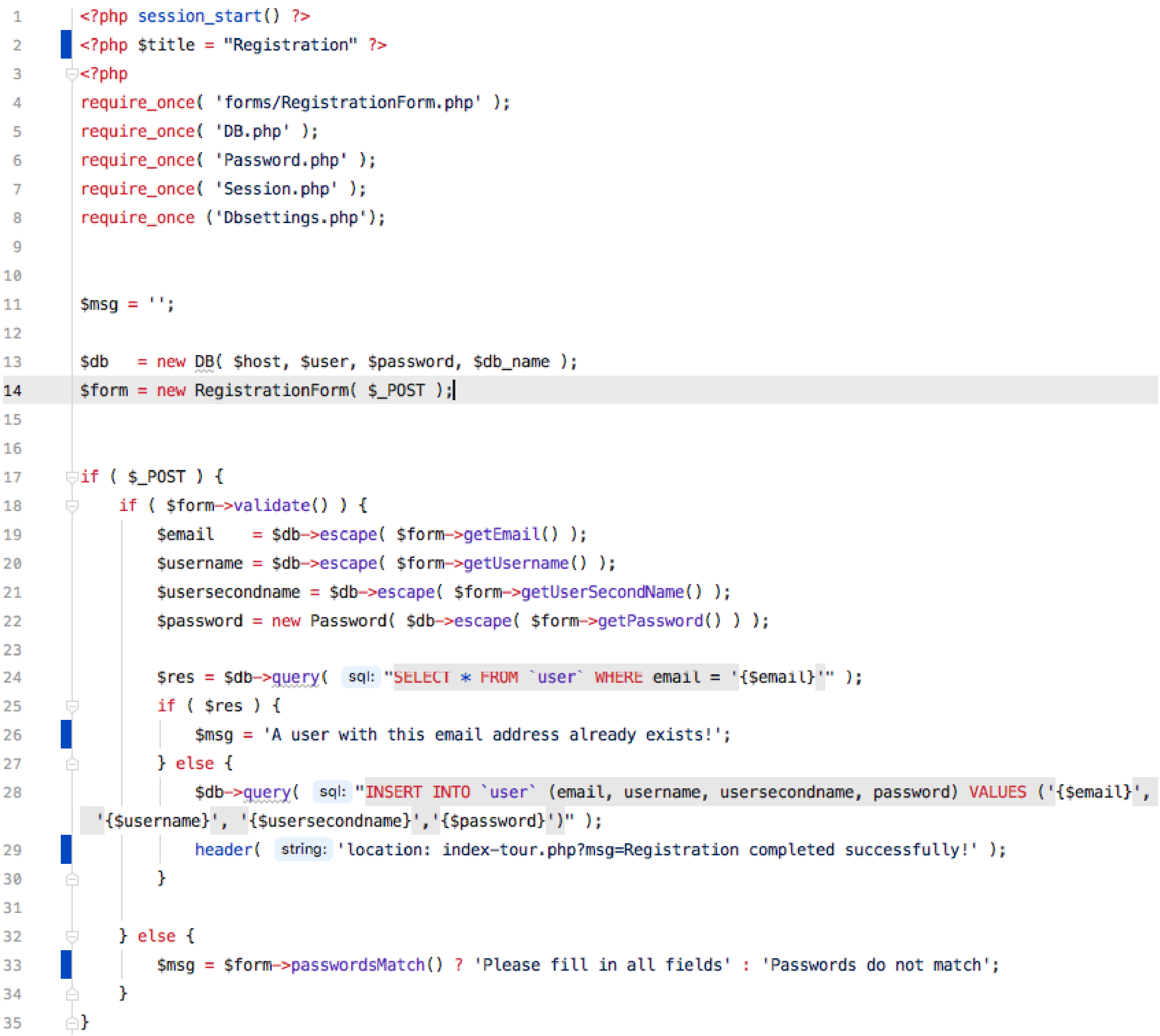
[48] Author’s desktop skreen



```
<?php include 'header.php' ?>
<h4 class="text-center">Registration</h4>
<b style="...">"<?=$msg; ?></b>
<form method="post">
    <div class="form-group">
        <label for="InputEmail">E-mail address</label>
        <input type="email" class="form-control" id="InputEmail" placeholder="Your email" name="email"
               value="<?= $form->getEmail(); ?>">
    </div>
    <div class="form-group">
        <label for="InputUsername">Username</label>
        <input type="text" class="form-control" id="InputUsername"
               placeholder="Username" name="username" value="<?= $form->getUsername() ?>">
    </div>
    <div class="form-group">
        <label for="InputUserSecondName">Last Name</label>
        <input type="text" class="form-control" id="InputUserSecondName" placeholder="Last Name"
               name="usersecondname" value="<?= $form->getUserSecondName() ?>">
    </div>
    <div class="form-group">
        <label for="InputPassword">Password</label>
        <input type="password" class="form-control" id="InputPassword" placeholder="Password" name="password">
    </div>
    <div class="form-group">
        <label for="InputPasswordConfirm">Password Verification</label>
        <input type="password" class="form-control" id="InputPasswordConfirm" placeholder="Password Verification"
               name="passwordConfirm">
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
    <a href="index-tour.php" class="btn btn-primary">Cancel</a>

</form>
<?php include 'footer.php' ?>
```

Picture 25 Registration form code[49]

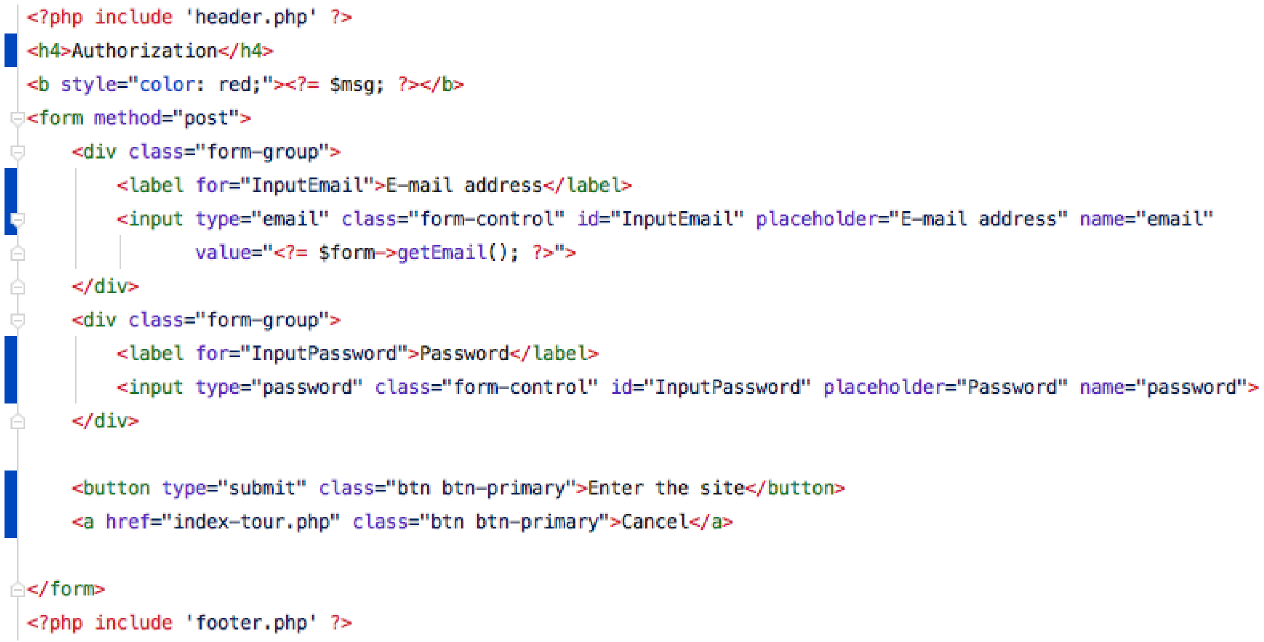
[49] Author's desktop skreen



```php
<?php session_start() ?>
<?php $title = "Registration" ?>
<?php
require_once( 'forms/RegistrationForm.php' );
require_once( 'DB.php' );
require_once( 'Password.php' );
require_once( 'Session.php' );
require_once ('Dbsettings.php');


$msg = '';

$db   = new DB( $host, $user, $password, $db_name );
$form = new RegistrationForm( $_POST );


if ( $_POST ) {
    if ( $form->validate() ) {
        $email    = $db->escape( $form->getEmail() );
        $username = $db->escape( $form->getUsername() );
        $usersecondname = $db->escape( $form->getUserSecondName() );
        $password = new Password( $db->escape( $form->getPassword() ) );

        $res = $db->query( "SELECT * FROM `user` WHERE email = '{$email}'" );
        if ( $res ) {
            $msg = 'A user with this email address already exists!';
        } else {
            $db->query( "INSERT INTO `user` (email, username, usersecondname, password) VALUES ('{$email}', '{$username}', '{$usersecondname}','{$password}')" );
            header( 'location: index-tour.php?msg=Registration completed successfully!' );
        }

    } else {
        $msg = $form->passwordsMatch() ? 'Please fill in all fields' : 'Passwords do not match';
    }
}
```

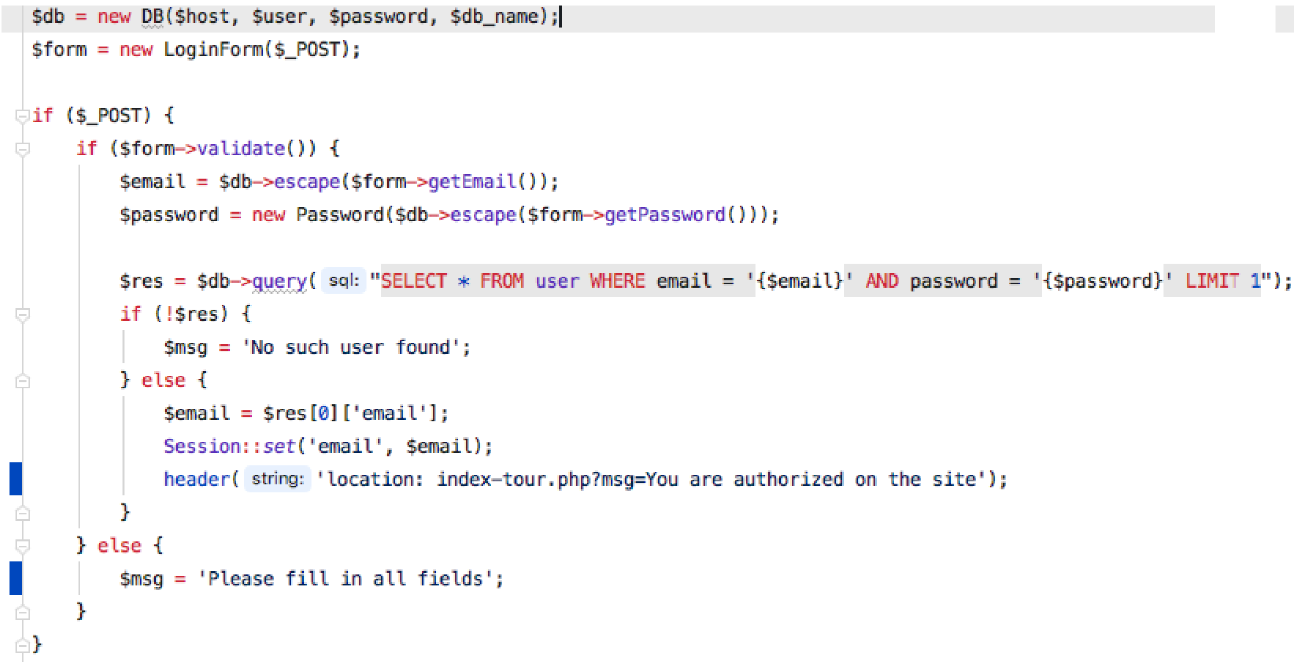
Picture 26 Registration to system code[50]

[50] Author’s desktop skreen



```
<?php include 'header.php' ?>
<h4>Authorization</h4>
<b style="color: red;"><?= $msg; ?></b>
<form method="post">
    <div class="form-group">
        <label for="InputEmail">E-mail address</label>
        <input type="email" class="form-control" id="InputEmail" placeholder="E-mail address" name="email"
               value="<?= $form->getEmail(); ?>">
    </div>
    <div class="form-group">
        <label for="InputPassword">Password</label>
        <input type="password" class="form-control" id="InputPassword" placeholder="Password" name="password">
    </div>

    <button type="submit" class="btn btn-primary">Enter the site</button>
    <a href="index-tour.php" class="btn btn-primary">Cancel</a>

</form>
<?php include 'footer.php' ?>
```

Picture 27Authorization form code[51]

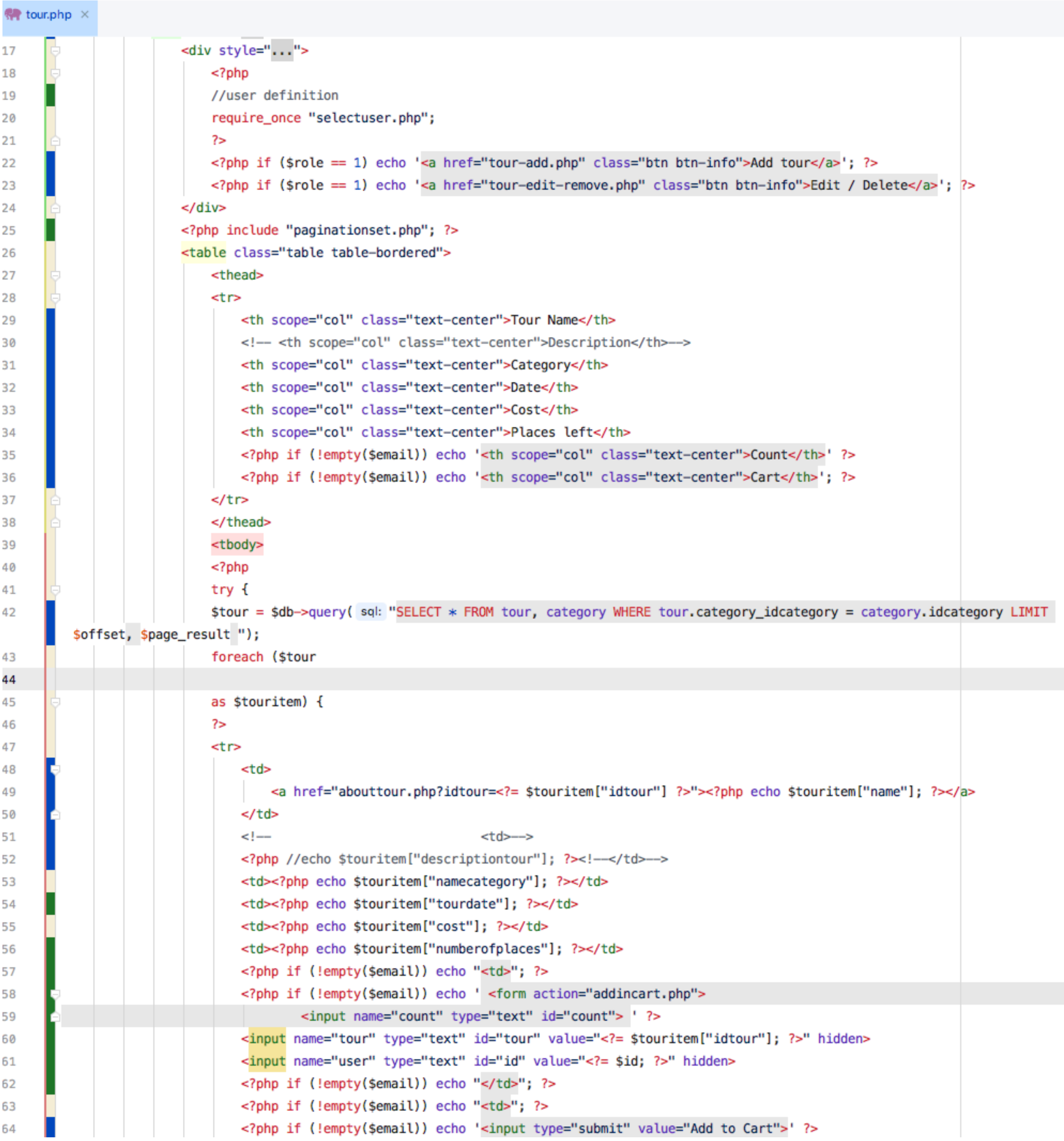


```
$db = new DB($host, $user, $password, $db_name);
$form = new LoginForm($_POST);

if ($_POST) {
    if ($form->validate()) {
        $email = $db->escape($form->getEmail());
        $password = new Password($db->escape($form->getPassword()));

        $res = $db->query( sql: "SELECT * FROM user WHERE email = '{$email}' AND password = '{$password}' LIMIT 1");
        if (!$res) {
            $msg = 'No such user found';
        } else {
            $email = $res[0]['email'];
            Session::set('email', $email);
            header( string: 'location: index-tour.php?msg=You are authorized on the site');
        }
    } else {
        $msg = 'Please fill in all fields';
    }
}
```

Picture 28 Authorization form processor’s code[52]

[51] Author’s desktop skreen

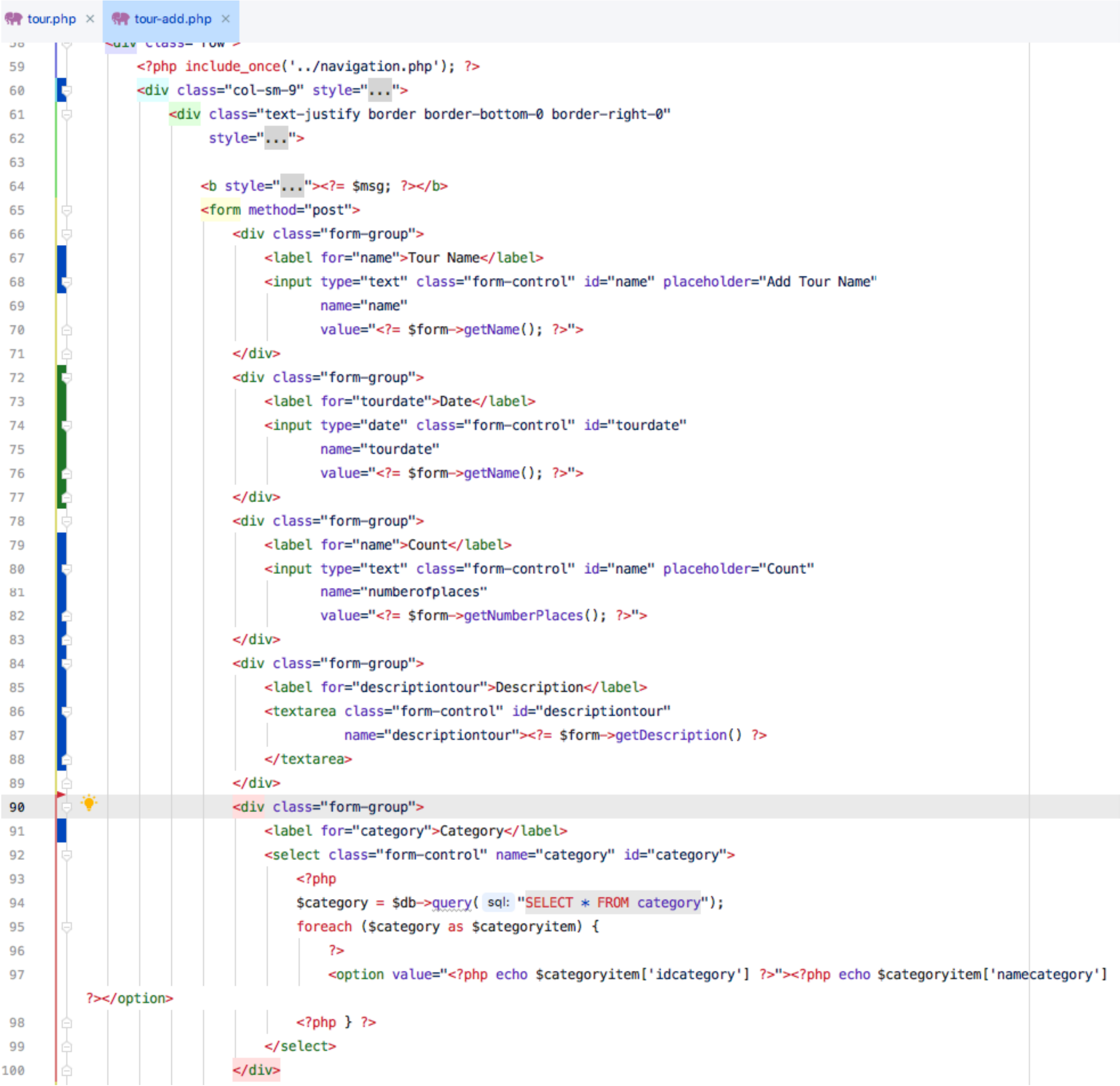
[52] Author’s desktop skreen



```
<div style="...">
    <?php
    //user definition
    require_once "selectuser.php";
    ?>
    <?php if ($role == 1) echo '<a href="tour-add.php" class="btn btn-info">Add tour</a>'; ?>
    <?php if ($role == 1) echo '<a href="tour-edit-remove.php" class="btn btn-info">Edit / Delete</a>'; ?>
</div>
<?php include "paginationset.php"; ?>
<table class="table table-bordered">
    <thead>
    <tr>
        <th scope="col" class="text-center">Tour Name</th>
        <!-- <th scope="col" class="text-center">Description</th>-->
        <th scope="col" class="text-center">Category</th>
        <th scope="col" class="text-center">Date</th>
        <th scope="col" class="text-center">Cost</th>
        <th scope="col" class="text-center">Places left</th>
        <?php if (!empty($email)) echo '<th scope="col" class="text-center">Count</th>' ?>
        <?php if (!empty($email)) echo '<th scope="col" class="text-center">Cart</th>'; ?>
    </tr>
    </thead>
    <tbody>
    <?php
    try {
    $tour = $db->query( sql: "SELECT * FROM tour, category WHERE tour.category_idcategory = category.idcategory LIMIT
$offset, $page_result ");
    foreach ($tour

    as $touritem) {
    ?>
    <tr>
        <td>
            <a href="abouttour.php?idtour=<?= $touritem["idtour"] ?>"><?php echo $touritem["name"]; ?></a>
        </td>
        <!--                            <td>-->
        <?php //echo $touritem["descriptiontour"]; ?><!--</td>-->
        <td><?php echo $touritem["namecategory"]; ?></td>
        <td><?php echo $touritem["tourdate"]; ?></td>
        <td><?php echo $touritem["cost"]; ?></td>
        <td><?php echo $touritem["numberofplaces"]; ?></td>
        <?php if (!empty($email)) echo "<td>"; ?>
        <?php if (!empty($email)) echo ' <form action="addincart.php">
                <input name="count" type="text" id="count"> ' ?>
        <input name="tour" type="text" id="tour" value="<?= $touritem["idtour"]; ?>" hidden>
        <input name="user" type="text" id="id" value="<?= $id; ?>" hidden>
        <?php if (!empty($email)) echo "</td>"; ?>
        <?php if (!empty($email)) echo "<td>"; ?>
        <?php if (!empty($email)) echo '<input type="submit" value="Add to Cart">' ?>
```

Picture 29 Tour list display table code[53]

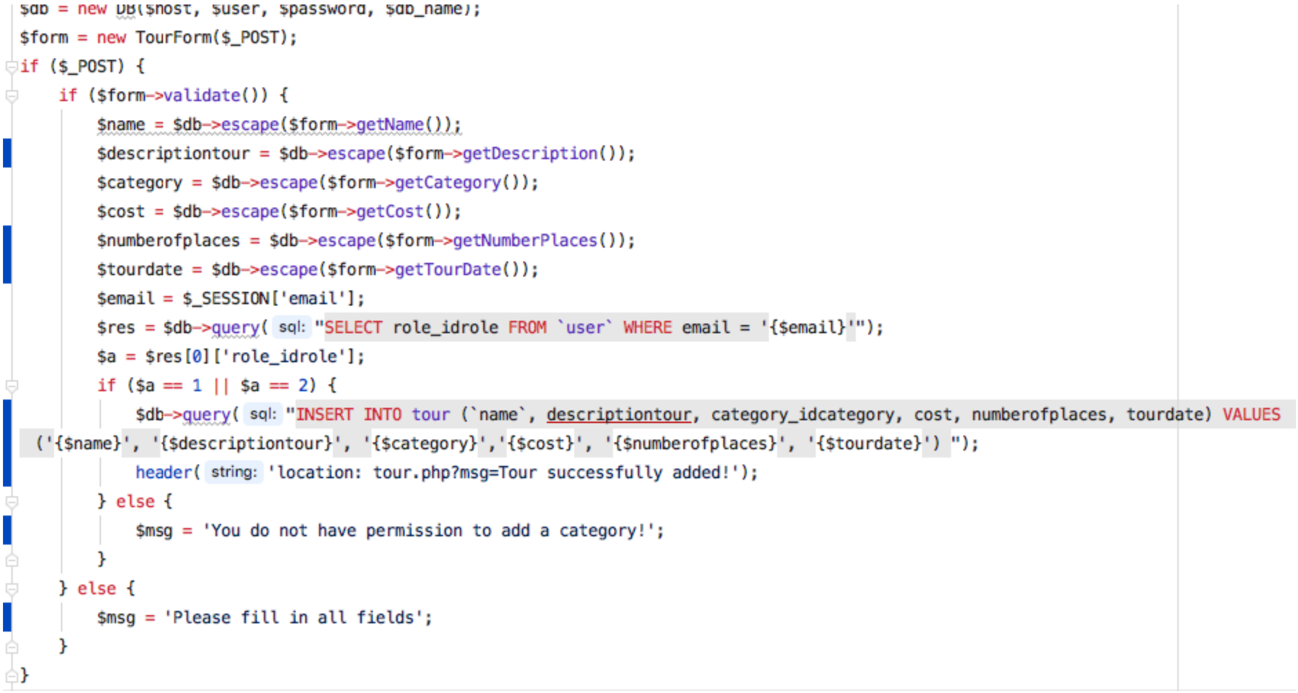
[53] Author’s desktop skreen



```
<div class="row">
    <?php include_once('../navigation.php'); ?>
    <div class="col-sm-9" style="...">
        <div class="text-justify border border-bottom-0 border-right-0"
             style="...">

            <b style="..."><?= $msg; ?></b>
            <form method="post">
                <div class="form-group">
                    <label for="name">Tour Name</label>
                    <input type="text" class="form-control" id="name" placeholder="Add Tour Name"
                           name="name"
                           value="<?= $form->getName(); ?>">
                </div>
                <div class="form-group">
                    <label for="tourdate">Date</label>
                    <input type="date" class="form-control" id="tourdate"
                           name="tourdate"
                           value="<?= $form->getName(); ?>">
                </div>
                <div class="form-group">
                    <label for="name">Count</label>
                    <input type="text" class="form-control" id="name" placeholder="Count"
                           name="numberofplaces"
                           value="<?= $form->getNumberPlaces(); ?>">
                </div>
                <div class="form-group">
                    <label for="descriptiontour">Description</label>
                    <textarea class="form-control" id="descriptiontour"
                              name="descriptiontour"><?= $form->getDescription() ?>
                    </textarea>
                </div>
                <div class="form-group">
                    <label for="category">Category</label>
                    <select class="form-control" name="category" id="category">
                        <?php
                        $category = $db->query( sql: "SELECT * FROM category");
                        foreach ($category as $categoryitem) {
                            ?>
                            <option value="<?php echo $categoryitem['idcategory'] ?>"><?php echo $categoryitem['namecategory']
?></option>
                        <?php } ?>
                    </select>
                </div>
```

Picture 30 Tour adding form code[54]

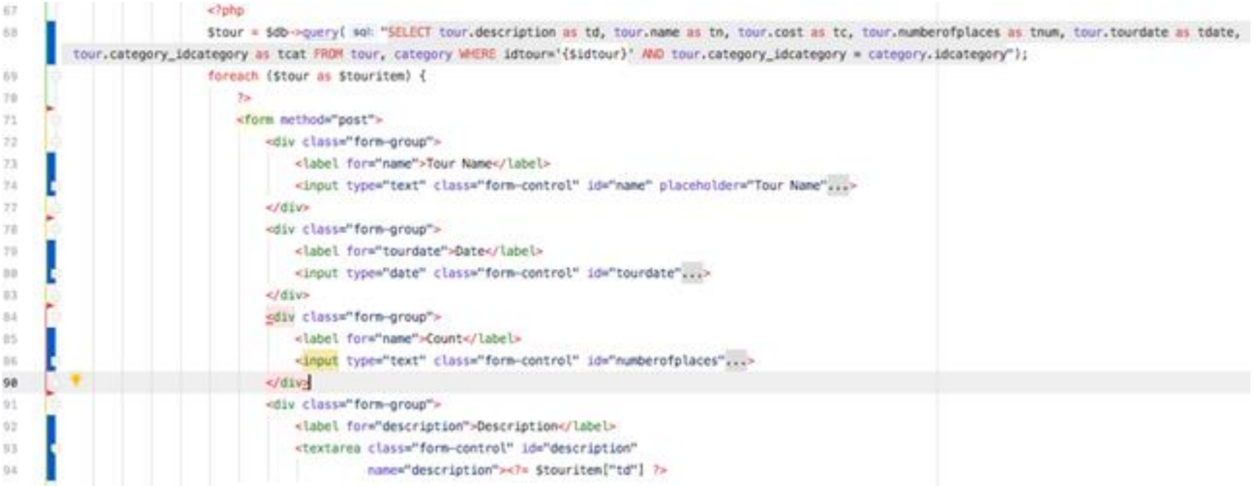
[54] Author’s desktop skreen



```
$db = new DB($host, $user, $password, $db_name);
$form = new TourForm($_POST);
if ($_POST) {
    if ($form->validate()) {
        $name = $db->escape($form->getName());
        $descriptiontour = $db->escape($form->getDescription());
        $category = $db->escape($form->getCategory());
        $cost = $db->escape($form->getCost());
        $numberofplaces = $db->escape($form->getNumberPlaces());
        $tourdate = $db->escape($form->getTourDate());
        $email = $_SESSION['email'];
        $res = $db->query( sql: "SELECT role_idrole FROM `user` WHERE email = '{$email}'");
        $a = $res[0]['role_idrole'];
        if ($a == 1 || $a == 2) {
            $db->query( sql: "INSERT INTO tour (`name`, descriptiontour, category_idcategory, cost, numberofplaces, tourdate) VALUES
  ('{$name}', '{$descriptiontour}', '{$category}','{$cost}', '{$numberofplaces}', '{$tourdate}') ");
            header( string: 'location: tour.php?msg=Tour successfully added!');
        } else {
            $msg = 'You do not have permission to add a category!';
        }
    } else {
        $msg = 'Please fill in all fields';
    }
}
```

Picture 31 Tour adding form processor’s code[55]

its processor (Picture 33).

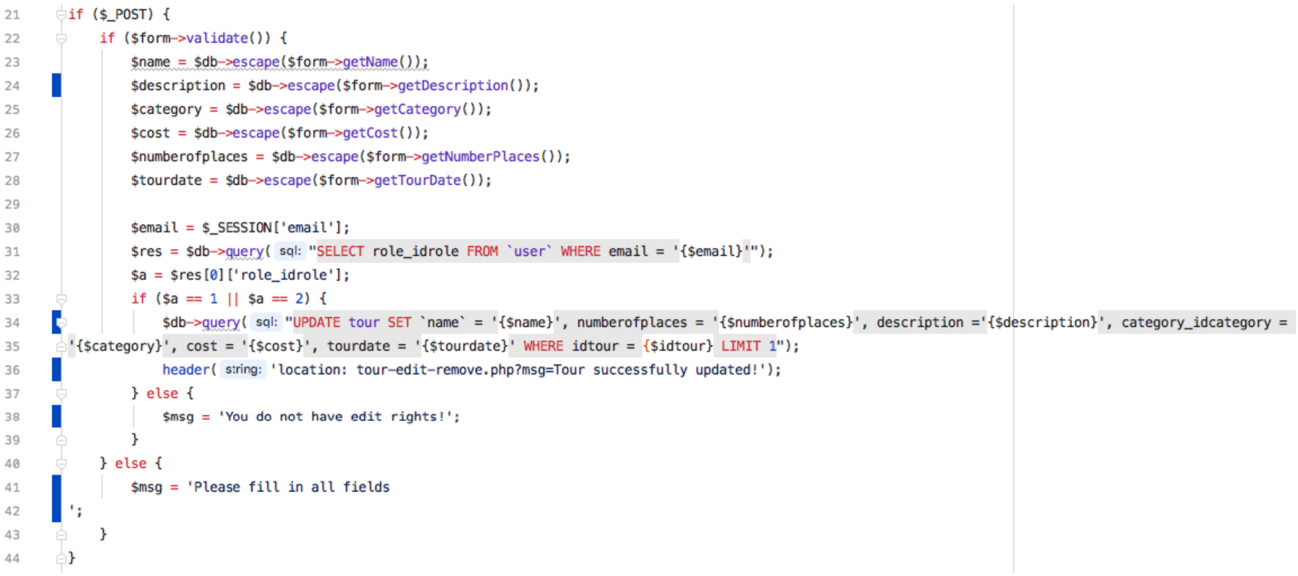


```
                  <?php
                  $tour = $db->query( sql: "SELECT tour.description as td, tour.name as tn, tour.cost as tc, tour.numberofplaces as tnum, tour.tourdate as tdate,
tour.category_idcategory as tcat FROM tour, category WHERE idtour='{$idtour}' AND tour.category_idcategory = category.idcategory");
                  foreach ($tour as $touritem) {
                      ?>
                      <form method="post">
                          <div class="form-group">
                              <label for="name">Tour Name</label>
                              <input type="text" class="form-control" id="name" placeholder="Tour Name"...>
                          </div>
                          <div class="form-group">
                              <label for="tourdate">Date</label>
                              <input type="date" class="form-control" id="tourdate"...>
                          </div>
                          <div class="form-group">
                              <label for="name">Count</label>
                              <input type="text" class="form-control" id="numberofplaces"...>
                          </div>
                          <div class="form-group">
                              <label for="description">Description</label>
                              <textarea class="form-control" id="description"
                                        name="description"><?= $touritem["td"] ?>
```

Picture 32 Tour editing form’s code[56]

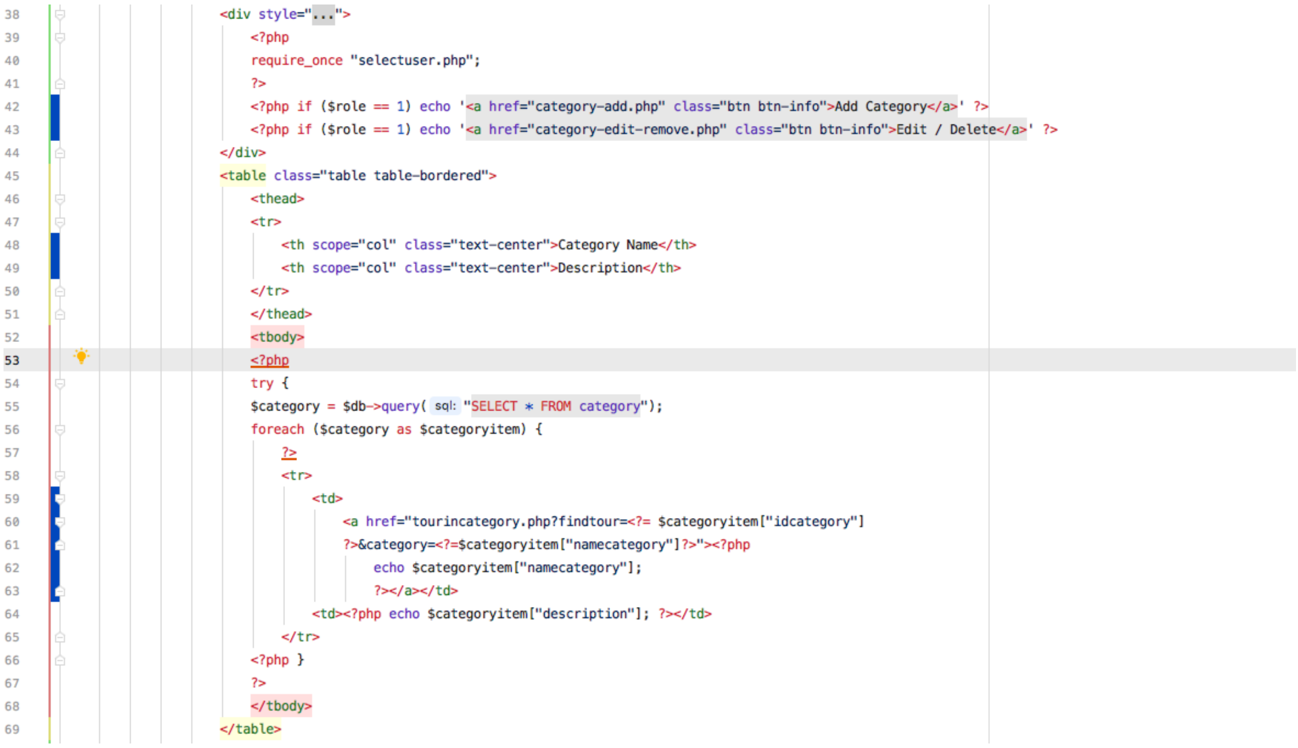
[55] Author’s desktop skreen

[56] Author’s desktop skreen



```
if ($_POST) {
    if ($form->validate()) {
        $name = $db->escape($form->getName());
        $description = $db->escape($form->getDescription());
        $category = $db->escape($form->getCategory());
        $cost = $db->escape($form->getCost());
        $numberofplaces = $db->escape($form->getNumberPlaces());
        $tourdate = $db->escape($form->getTourDate());

        $email = $_SESSION['email'];
        $res = $db->query( sql: "SELECT role_idrole FROM `user` WHERE email = '{$email}'");
        $a = $res[0]['role_idrole'];
        if ($a == 1 || $a == 2) {
            $db->query( sql: "UPDATE tour SET `name` = '{$name}', numberofplaces = '{$numberofplaces}', description ='{$description}', category_idcategory =
'{$category}', cost = '{$cost}', tourdate = '{$tourdate}' WHERE idtour = {$idtour} LIMIT 1");
            header( string: 'location: tour-edit-remove.php?msg=Tour successfully updated!');
        } else {
            $msg = 'You do not have edit rights!';
        }
    } else {
        $msg = 'Please fill in all fields
';
    }
}
```

Picture 33 Tours editing form processor's code[57]

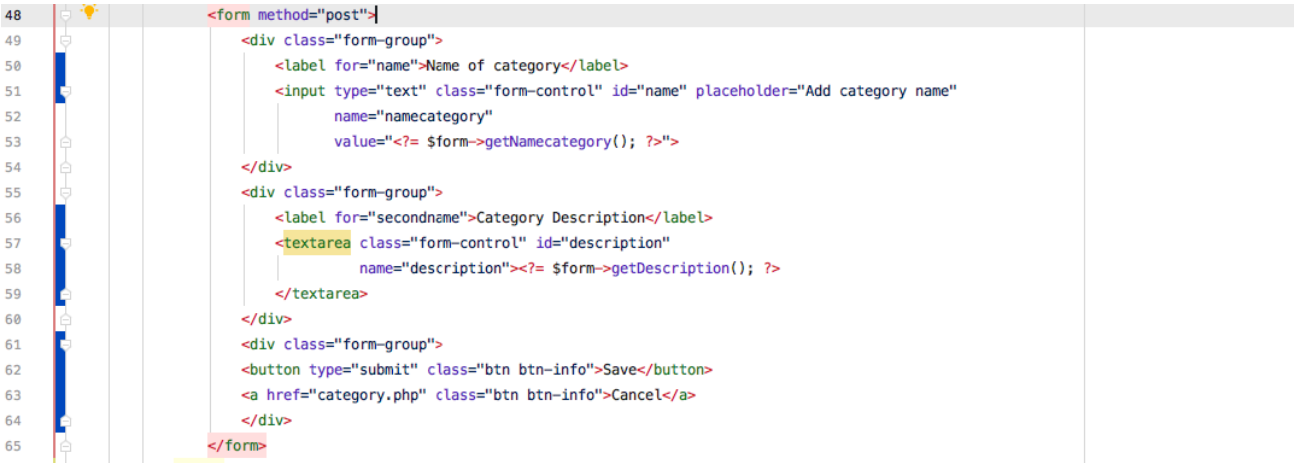


```
<div style="...">
    <?php
    require_once "selectuser.php";
    ?>
    <?php if ($role == 1) echo '<a href="category-add.php" class="btn btn-info">Add Category</a>' ?>
    <?php if ($role == 1) echo '<a href="category-edit-remove.php" class="btn btn-info">Edit / Delete</a>' ?>
</div>
<table class="table table-bordered">
    <thead>
    <tr>
        <th scope="col" class="text-center">Category Name</th>
        <th scope="col" class="text-center">Description</th>
    </tr>
    </thead>
    <tbody>
    <?php
    try {
    $category = $db->query( sql: "SELECT * FROM category");
    foreach ($category as $categoryitem) {
        ?>
        <tr>
            <td>
                <a href="tourincategory.php?findtour=<?= $categoryitem["idcategory"]
                ?>&category=<?=$categoryitem["namecategory"]?>"><?php
                    echo $categoryitem["namecategory"];
                    ?></a></td>
            <td><?php echo $categoryitem["description"]; ?></td>
        </tr>
    <?php }
    ?>
    </tbody>
</table>
```

Picture 34 Tour category list's display table code[58]

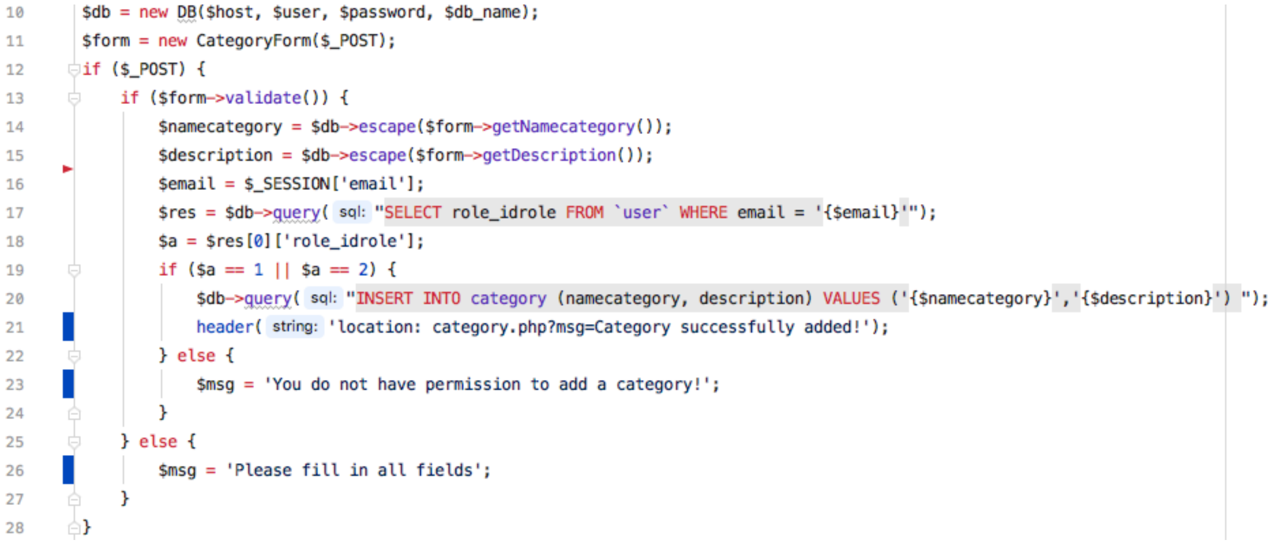
[57] Author's desktop skreen

[58] Author's desktop skreen



```
<form method="post">
    <div class="form-group">
        <label for="name">Name of category</label>
        <input type="text" class="form-control" id="name" placeholder="Add category name"
               name="namecategory"
               value="<?= $form->getNamecategory(); ?>">
    </div>
    <div class="form-group">
        <label for="secondname">Category Description</label>
        <textarea class="form-control" id="description"
                  name="description"><?= $form->getDescription(); ?>
        </textarea>
    </div>
    <div class="form-group">
    <button type="submit" class="btn btn-info">Save</button>
    <a href="category.php" class="btn btn-info">Cancel</a>
    </div>
</form>
```

Picture 35 Category adding form’s code[59]

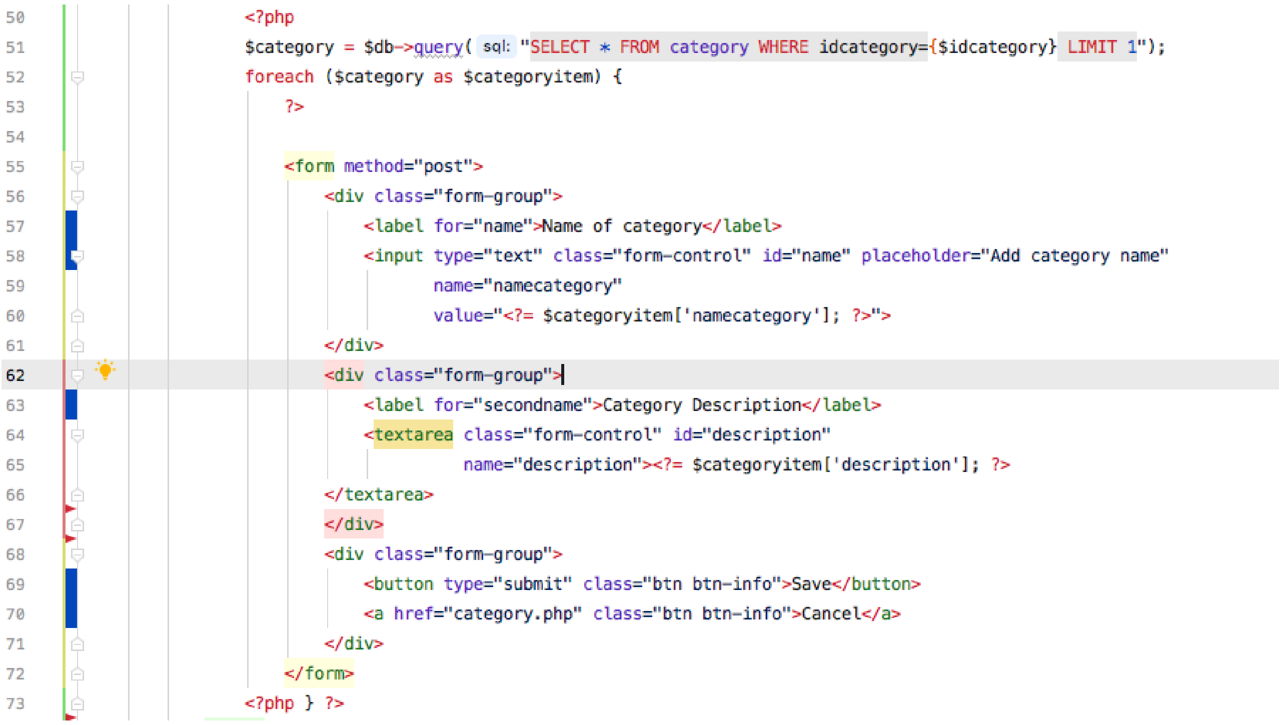


```
$db = new DB($host, $user, $password, $db_name);
$form = new CategoryForm($_POST);
if ($_POST) {
    if ($form->validate()) {
        $namecategory = $db->escape($form->getNamecategory());
        $description = $db->escape($form->getDescription());
        $email = $_SESSION['email'];
        $res = $db->query( sql: "SELECT role_idrole FROM `user` WHERE email = '{$email}'");
        $a = $res[0]['role_idrole'];
        if ($a == 1 || $a == 2) {
            $db->query( sql: "INSERT INTO category (namecategory, description) VALUES ('{$namecategory}','{$description}') ");
            header( string: 'location: category.php?msg=Category successfully added!');
        } else {
            $msg = 'You do not have permission to add a category!';
        }
    } else {
        $msg = 'Please fill in all fields';
    }
}
```

Picture 36 Category adding form’s processor code[60]

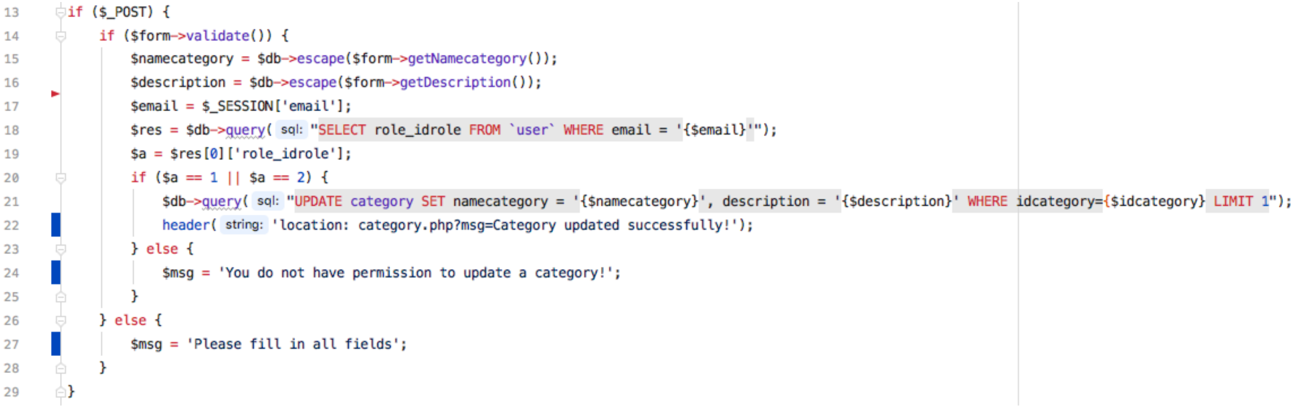
[59] Author’s desktop skreen

[60] Author’s desktop skreen



```
<?php
$category = $db->query("SELECT * FROM category WHERE idcategory={$idcategory} LIMIT 1");
foreach ($category as $categoryitem) {
    ?>

    <form method="post">
        <div class="form-group">
            <label for="name">Name of category</label>
            <input type="text" class="form-control" id="name" placeholder="Add category name"
                   name="namecategory"
                   value="<?= $categoryitem['namecategory']; ?>">
        </div>
        <div class="form-group">
            <label for="secondname">Category Description</label>
            <textarea class="form-control" id="description"
                      name="description"><?= $categoryitem['description']; ?>
        </textarea>
        </div>
        <div class="form-group">
            <button type="submit" class="btn btn-info">Save</button>
            <a href="category.php" class="btn btn-info">Cancel</a>
        </div>
    </form>
<?php } ?>
```

Picture 37 Category editing form’s code[61]

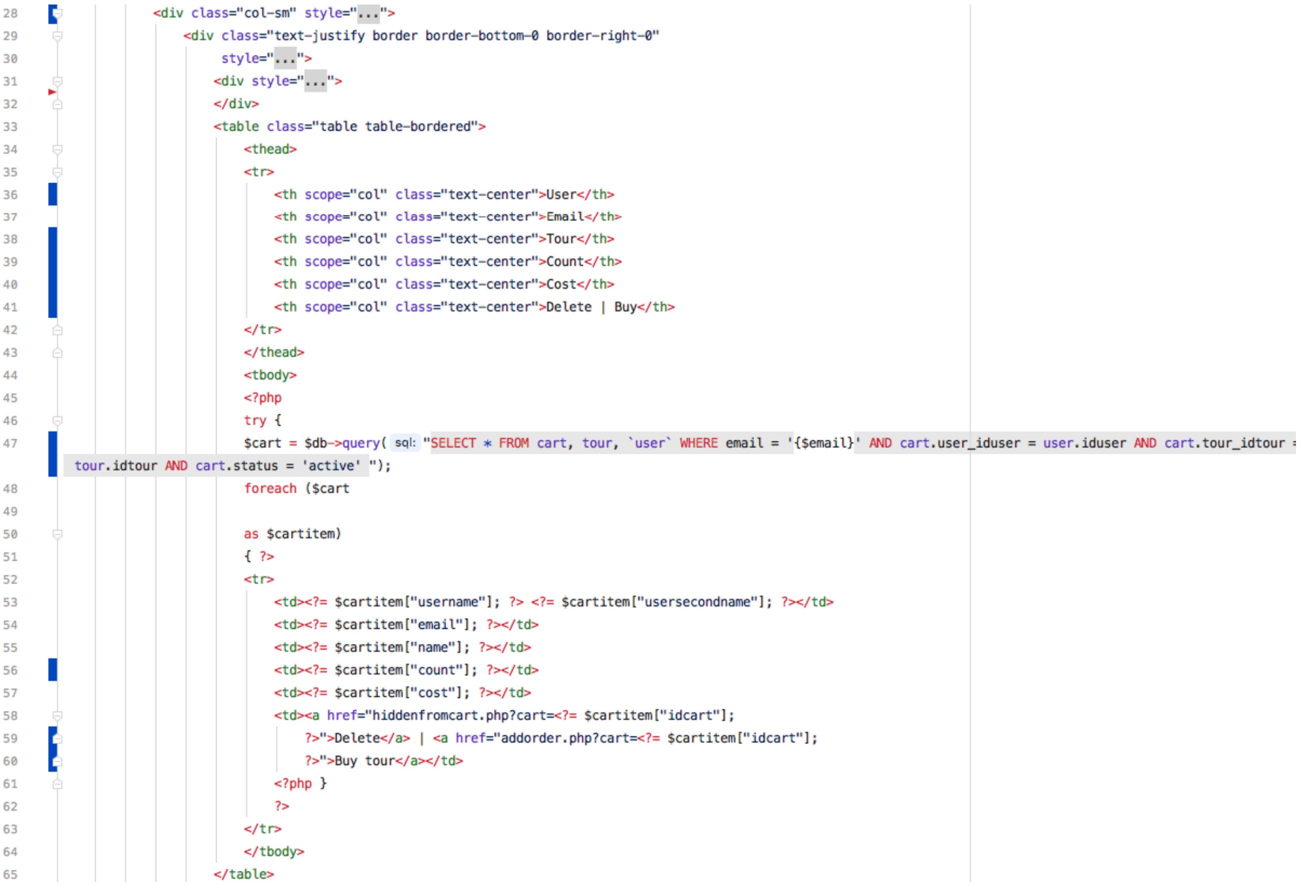


```
if ($_POST) {
    if ($form->validate()) {
        $namecategory = $db->escape($form->getNamecategory());
        $description = $db->escape($form->getDescription());
        $email = $_SESSION['email'];
        $res = $db->query("SELECT role_idrole FROM `user` WHERE email = '{$email}'");
        $a = $res[0]['role_idrole'];
        if ($a == 1 || $a == 2) {
            $db->query("UPDATE category SET namecategory = '{$namecategory}', description = '{$description}' WHERE idcategory={$idcategory} LIMIT 1");
            header('location: category.php?msg=Category updated successfully!');
        } else {
            $msg = 'You do not have permission to update a category!';
        }
    } else {
        $msg = 'Please fill in all fields';
    }
}
```

Picture 38 Category editing form’s processor code[62]

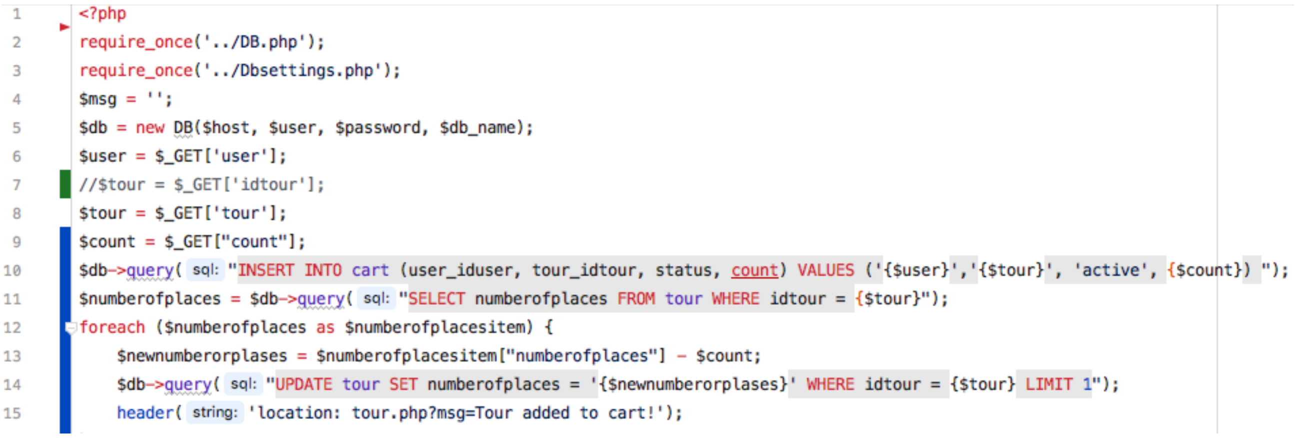
[61] Author’s desktop skreen

[62] Author’s desktop skreen



```
<div class="col-sm" style="...">
    <div class="text-justify border border-bottom-0 border-right-0"
         style="...">
        <div style="...">
        </div>
        <table class="table table-bordered">
            <thead>
            <tr>
                <th scope="col" class="text-center">User</th>
                <th scope="col" class="text-center">Email</th>
                <th scope="col" class="text-center">Tour</th>
                <th scope="col" class="text-center">Count</th>
                <th scope="col" class="text-center">Cost</th>
                <th scope="col" class="text-center">Delete | Buy</th>
            </tr>
            </thead>
            <tbody>
            <?php
            try {
            $cart = $db->query( sql: "SELECT * FROM cart, tour, `user` WHERE email = '{$email}' AND cart.user_iduser = user.iduser AND cart.tour_idtour =
 tour.idtour AND cart.status = 'active' ");
            foreach ($cart

            as $cartitem)
            { ?>
            <tr>
                <td><?= $cartitem["username"]; ?> <?= $cartitem["usersecondname"]; ?></td>
                <td><?= $cartitem["email"]; ?></td>
                <td><?= $cartitem["name"]; ?></td>
                <td><?= $cartitem["count"]; ?></td>
                <td><?= $cartitem["cost"]; ?></td>
                <td><a href="hiddenfromcart.php?cart=<?= $cartitem["idcart"];
                    ?>">Delete</a> | <a href="addorder.php?cart=<?= $cartitem["idcart"];
                    ?>">Buy tour</a></td>
                <?php }
                ?>
            </tr>
            </tbody>
        </table>
```

Picture 39 Data added to cart by user code[63]

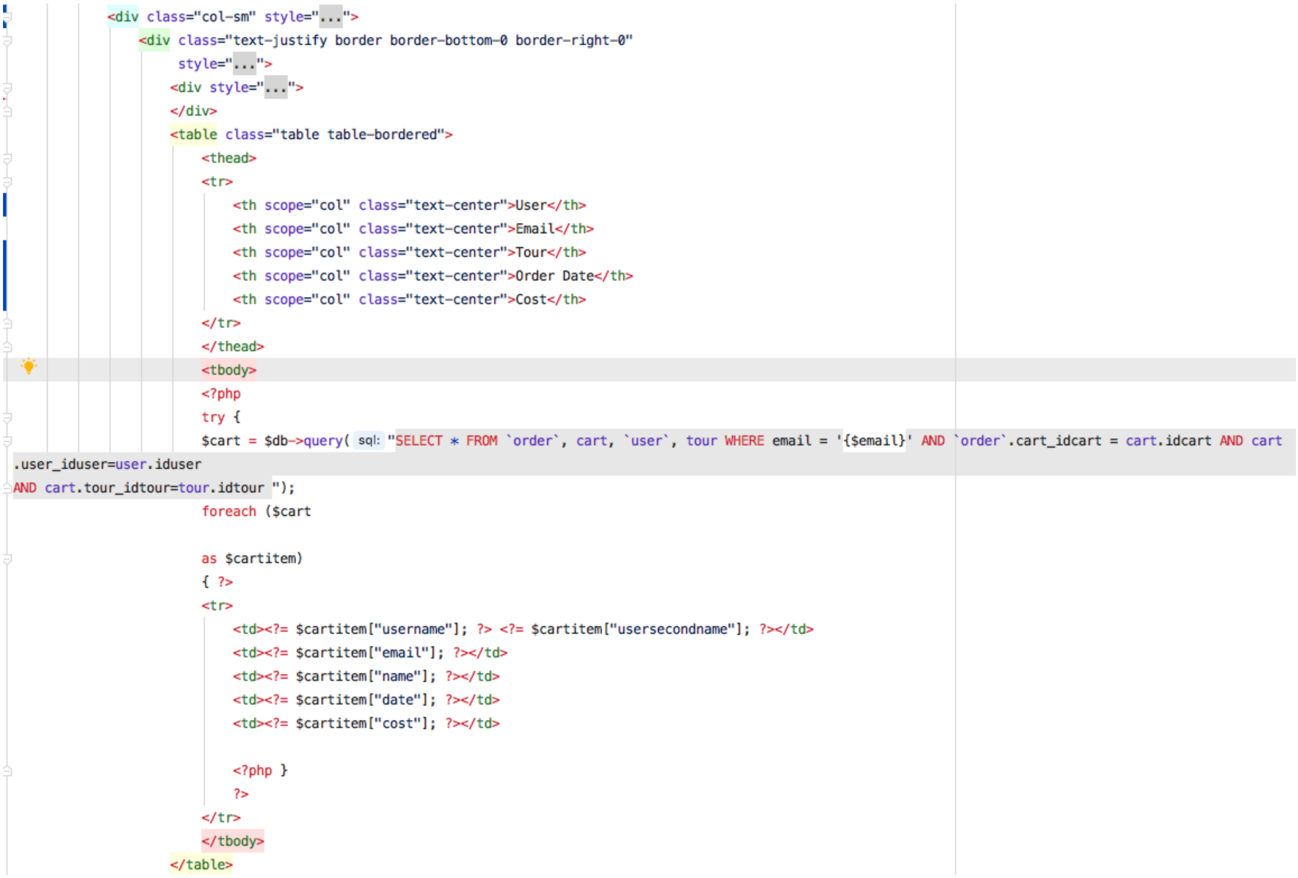


```
<?php
require_once('../DB.php');
require_once('../Dbsettings.php');
$msg = '';
$db = new DB($host, $user, $password, $db_name);
$user = $_GET['user'];
//$tour = $_GET['idtour'];
$tour = $_GET['tour'];
$count = $_GET["count"];
$db->query( sql: "INSERT INTO cart (user_iduser, tour_idtour, status, count) VALUES ('{$user}','{$tour}', 'active', {$count}) ");
$numberofplaces = $db->query( sql: "SELECT numberofplaces FROM tour WHERE idtour = {$tour}");
foreach ($numberofplaces as $numberofplacesitem) {
    $newnumberorplases = $numberofplacesitem["numberofplaces"] - $count;
    $db->query( sql: "UPDATE tour SET numberofplaces = '{$newnumberorplases}' WHERE idtour = {$tour} LIMIT 1");
    header( string: 'location: tour.php?msg=Tour added to cart!');
```

Picture 40 File with tour added to cart updated information[64]

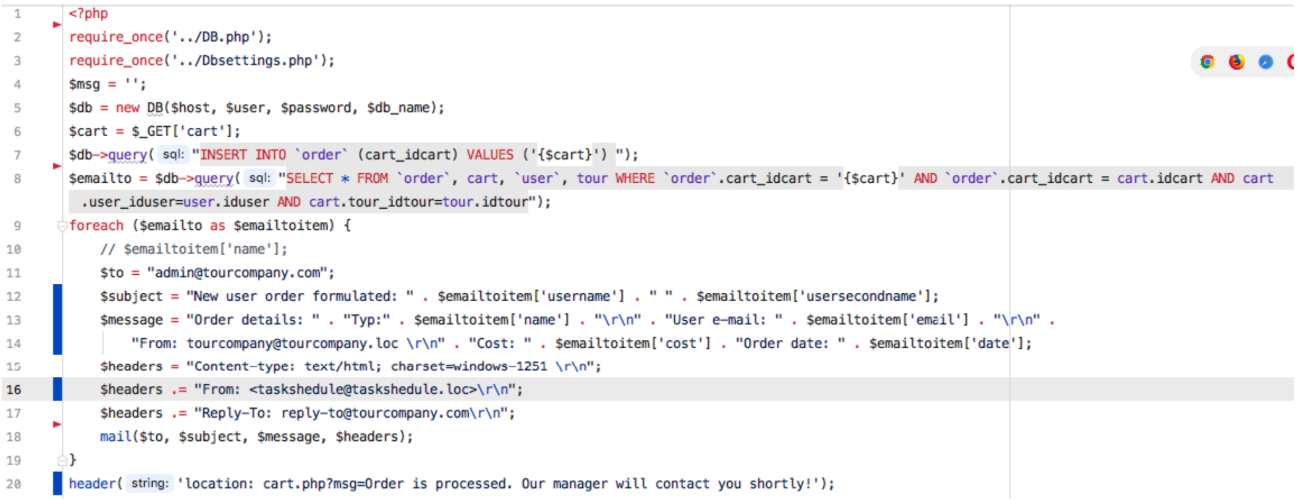
[63] Author's desktop skreen

[64] Author's desktop skreen



```
            <div class="col-sm" style="...">
                <div class="text-justify border border-bottom-0 border-right-0"
                     style="...">
                    <div style="...">
                    </div>
                    <table class="table table-bordered">
                        <thead>
                        <tr>
                            <th scope="col" class="text-center">User</th>
                            <th scope="col" class="text-center">Email</th>
                            <th scope="col" class="text-center">Tour</th>
                            <th scope="col" class="text-center">Order Date</th>
                            <th scope="col" class="text-center">Cost</th>
                        </tr>
                        </thead>
                        <tbody>
                        <?php
                        try {
                        $cart = $db->query( sql: "SELECT * FROM `order`, cart, `user`, tour WHERE email = '{$email}' AND `order`.cart_idcart = cart.idcart AND cart
.user_iduser=user.iduser
AND cart.tour_idtour=tour.idtour ");
                        foreach ($cart

                        as $cartitem)
                        { ?>
                        <tr>
                            <td><?= $cartitem["username"]; ?> <?= $cartitem["usersecondname"]; ?></td>
                            <td><?= $cartitem["email"]; ?></td>
                            <td><?= $cartitem["name"]; ?></td>
                            <td><?= $cartitem["date"]; ?></td>
                            <td><?= $cartitem["cost"]; ?></td>

                            <?php }
                            ?>
                        </tr>
                        </tbody>
                    </table>
```

Picture 41 Code for file with the confimed order from cart[65]

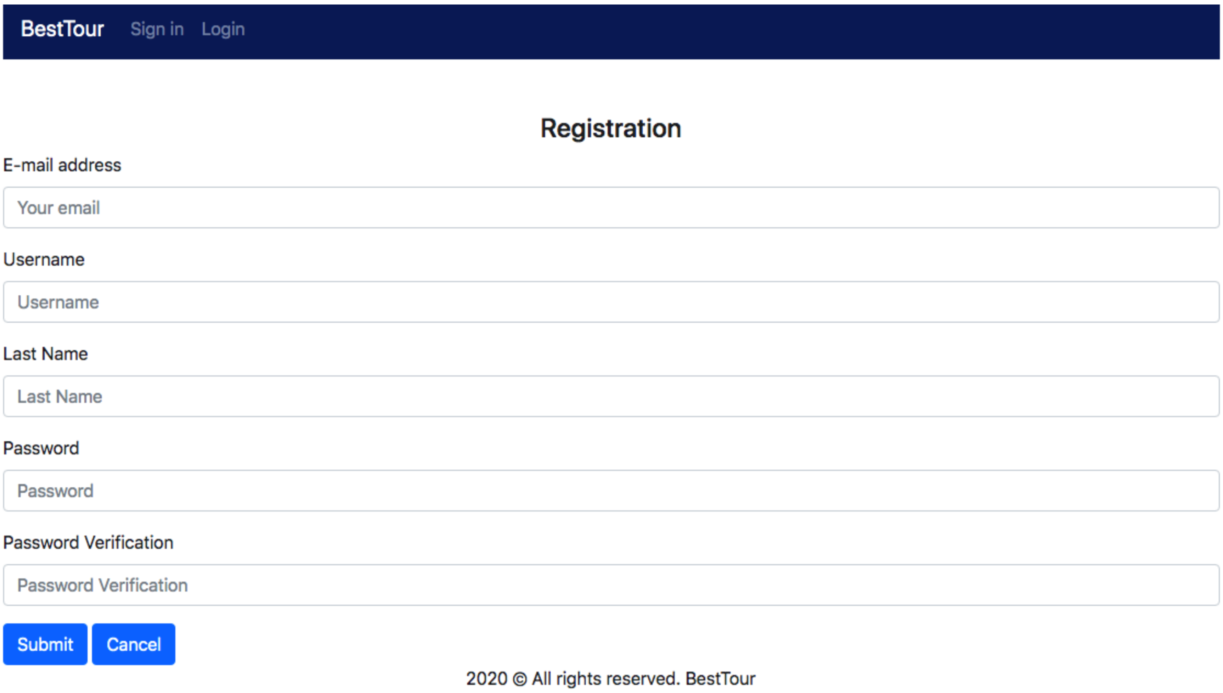


```
<?php
require_once('../DB.php');
require_once('../Dbsettings.php');
$msg = '';
$db = new DB($host, $user, $password, $db_name);
$cart = $_GET['cart'];
$db->query( sql: "INSERT INTO `order` (cart_idcart) VALUES ('{$cart}') ");
$emailto = $db->query( sql: "SELECT * FROM `order`, cart, `user`, tour WHERE `order`.cart_idcart = '{$cart}' AND `order`.cart_idcart = cart.idcart AND cart
 .user_iduser=user.iduser AND cart.tour_idtour=tour.idtour");
foreach ($emailto as $emailtoitem) {
    // $emailtoitem['name'];
    $to = "admin@tourcompany.com";
    $subject = "New user order formulated: " . $emailtoitem['username'] . " " . $emailtoitem['usersecondname'];
    $message = "Order details: " . "Typ:" . $emailtoitem['name'] . "\r\n" . "User e-mail: " . $emailtoitem['email'] . "\r\n" .
        "From: tourcompany@tourcompany.loc \r\n" . "Cost: " . $emailtoitem['cost'] . "Order date: " . $emailtoitem['date'];
    $headers = "Content-type: text/html; charset=windows-1251 \r\n";
    $headers .= "From: <taskshedule@taskshedule.loc>\r\n";
    $headers .= "Reply-To: reply-to@tourcompany.com\r\n";
    mail($to, $subject, $message, $headers);
}
header( string: 'location: cart.php?msg=Order is processed. Our manager will contact you shortly!');
```

Picture 42 File adding the orde to data base code [66]

[65] Author's desktop skreen

[66] Author's desktop skreen



Registration

E-mail address

Your email

Username

Username

Last Name

Last Name

Password

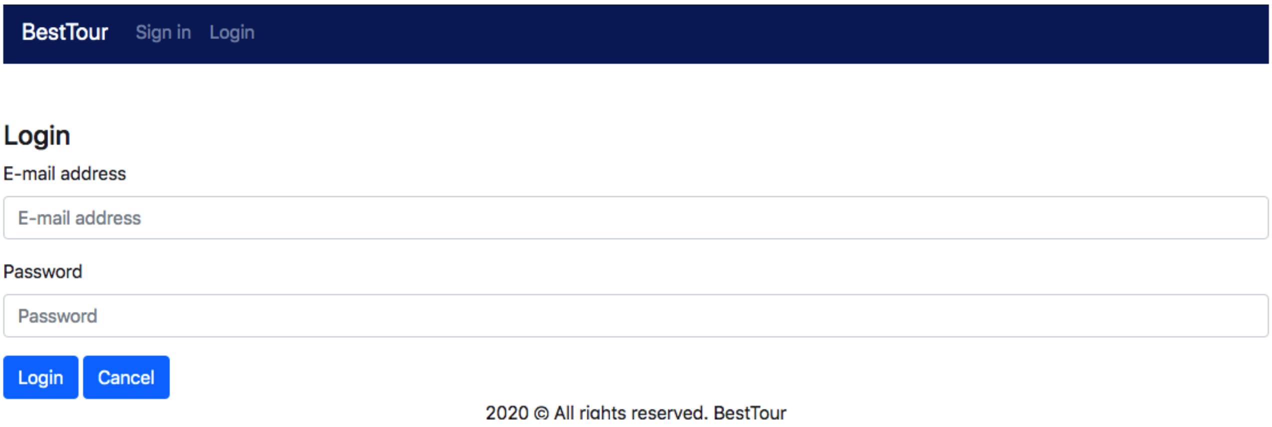
Password

Password Verification

Password Verification

2020 © All rights reserved. BestTour

Picture 43 Registration page[67]



Login

E-mail address

E-mail address

Password

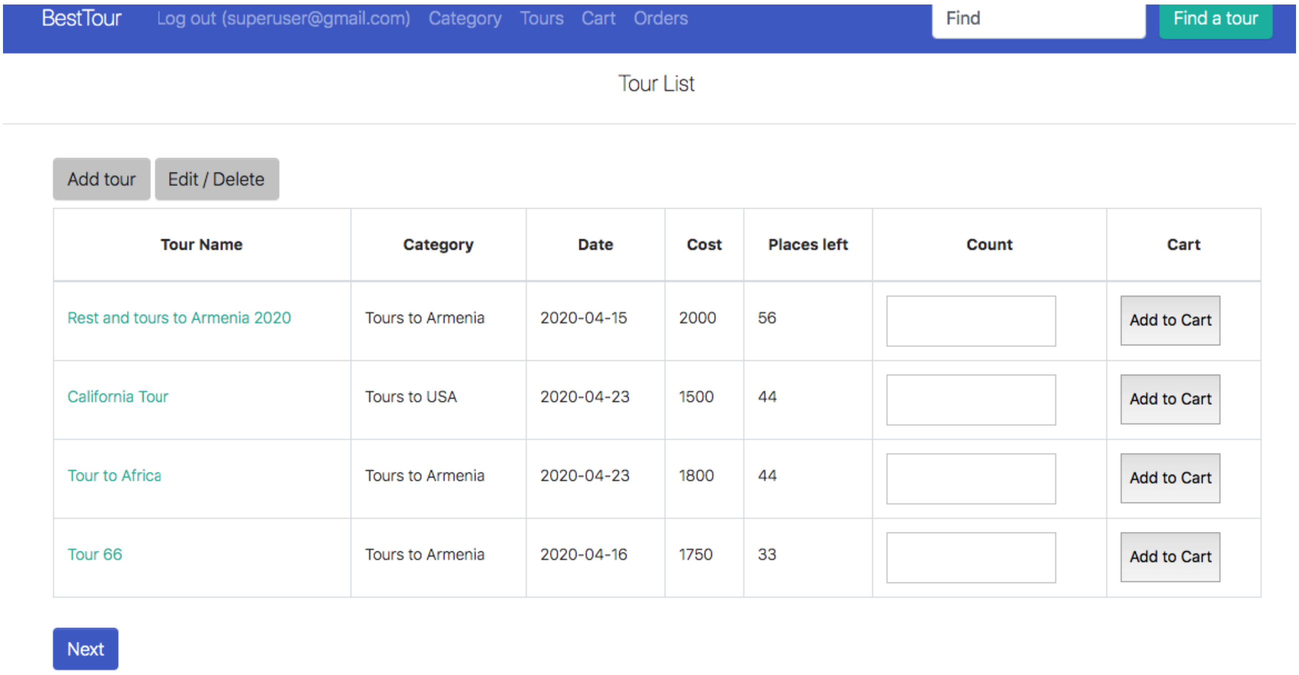
Password

2020 © All rights reserved. BestTour

Picture 44 Authorization page[68]

[67] Author’s desktop skreen

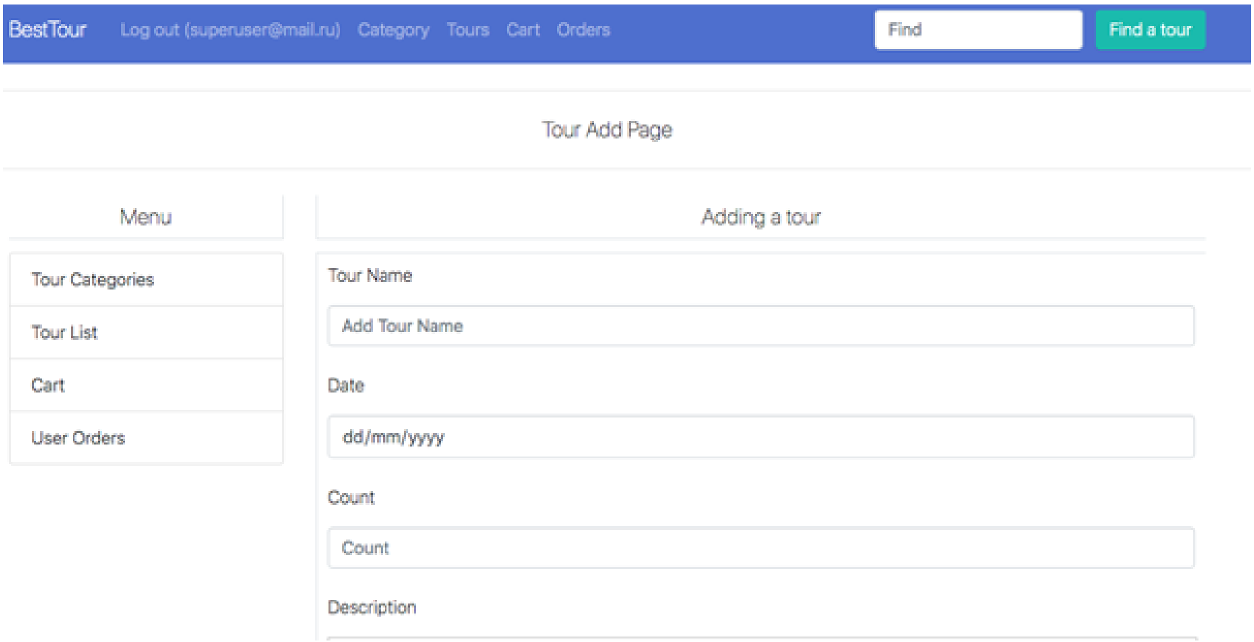
[68] Author’s desktop skreen



Tour List

Tour Name	Category	Date	Cost	Places left	Count	Cart
Rest and tours to Armenia 2020	Tours to Armenia	2020-04-15	2000	56		Add to Cart
California Tour	Tours to USA	2020-04-23	1500	44		Add to Cart
Tour to Africa	Tours to Armenia	2020-04-23	1800	44		Add to Cart
Tour 66	Tours to Armenia	2020-04-16	1750	33		Add to Cart

Picture 45 Tour list page[69]



Tour Add Page

Adding a tour

Tour Name

Add Tour Name

Date

dd/mm/yyyy

Count

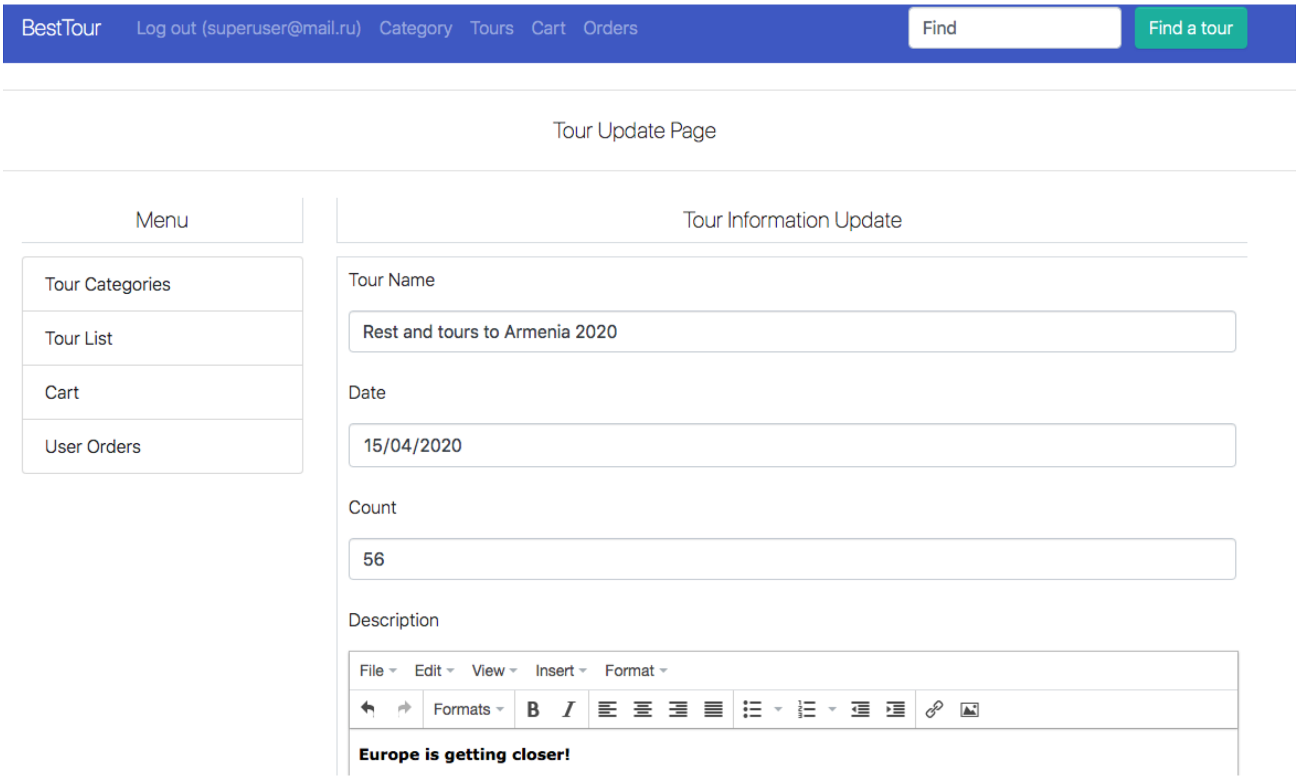
Count

Description

Picture 46 Tour adding page[70]

[69] Author’s desktop skreen

[70] Author’s desktop skreen



Tour Update Page

Tour Information Update

Tour Name: Rest and tours to Armenia 2020

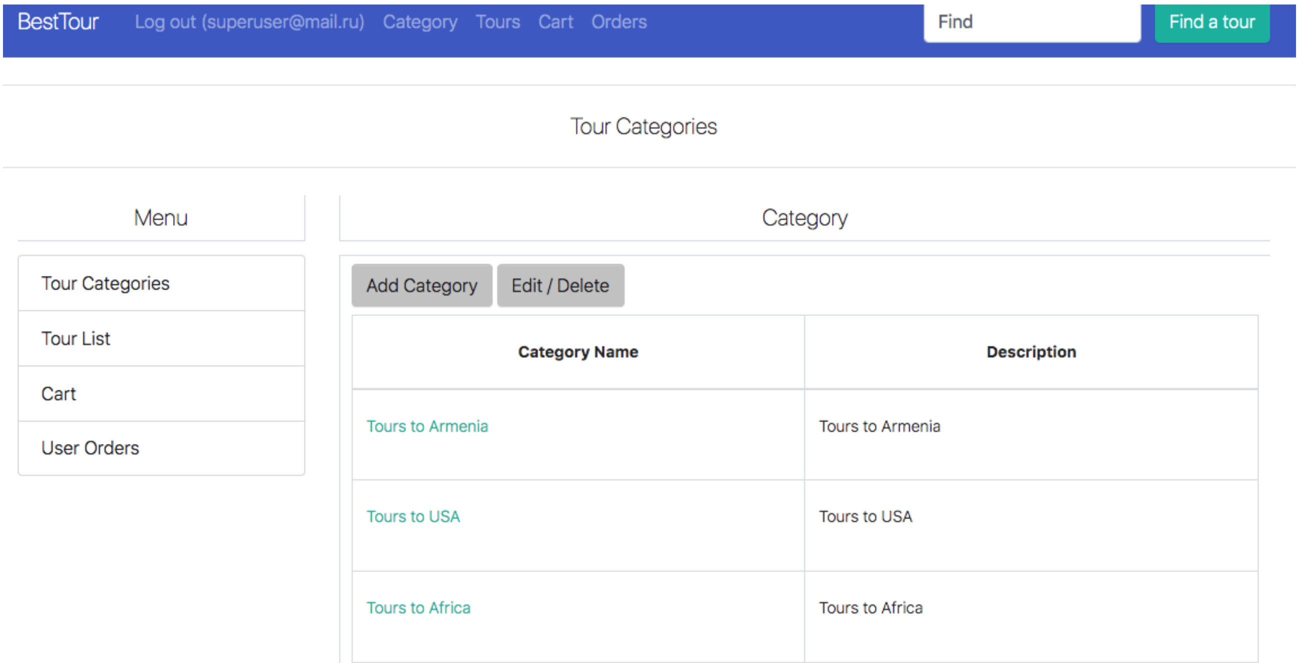
Date: 15/04/2020

Count: 56

Description

Europe is getting closer!

Picture 47 Page updated[71]



Tour Categories

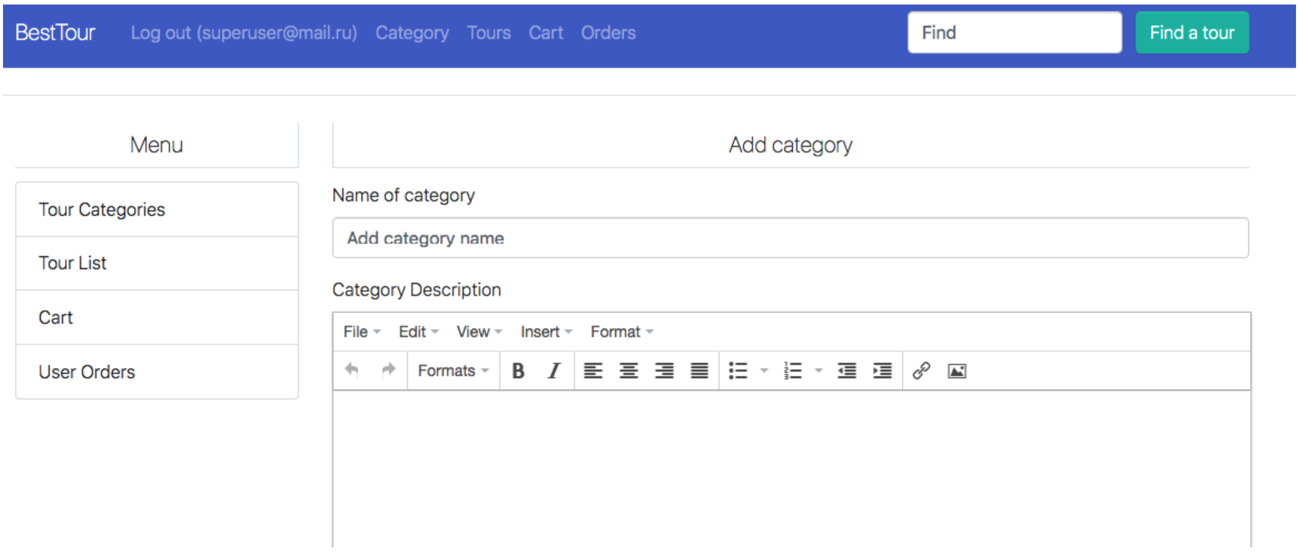
Category

Category Name	Description
Tours to Armenia	Tours to Armenia
Tours to USA	Tours to USA
Tours to Africa	Tours to Africa

Picture 48 Category list page[72]

[71] Author’s desktop skreen

[72] Author’s desktop skreen



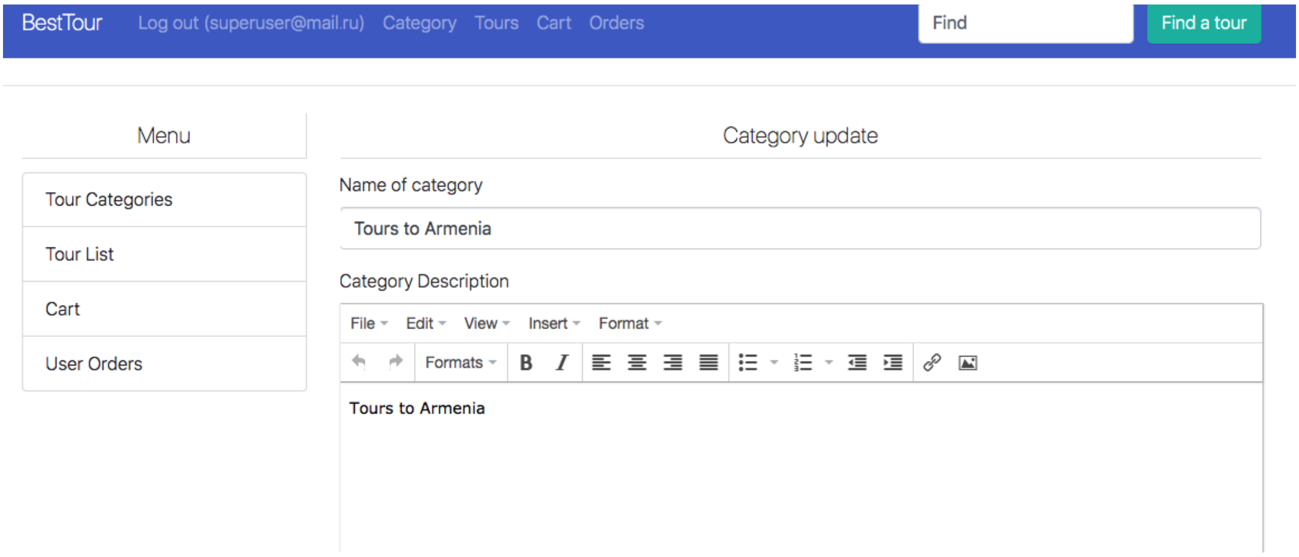
Add category

Name of category

Add category name

Category Description

Picture 49 Category adding page[73]



Category update

Name of category

Tours to Armenia

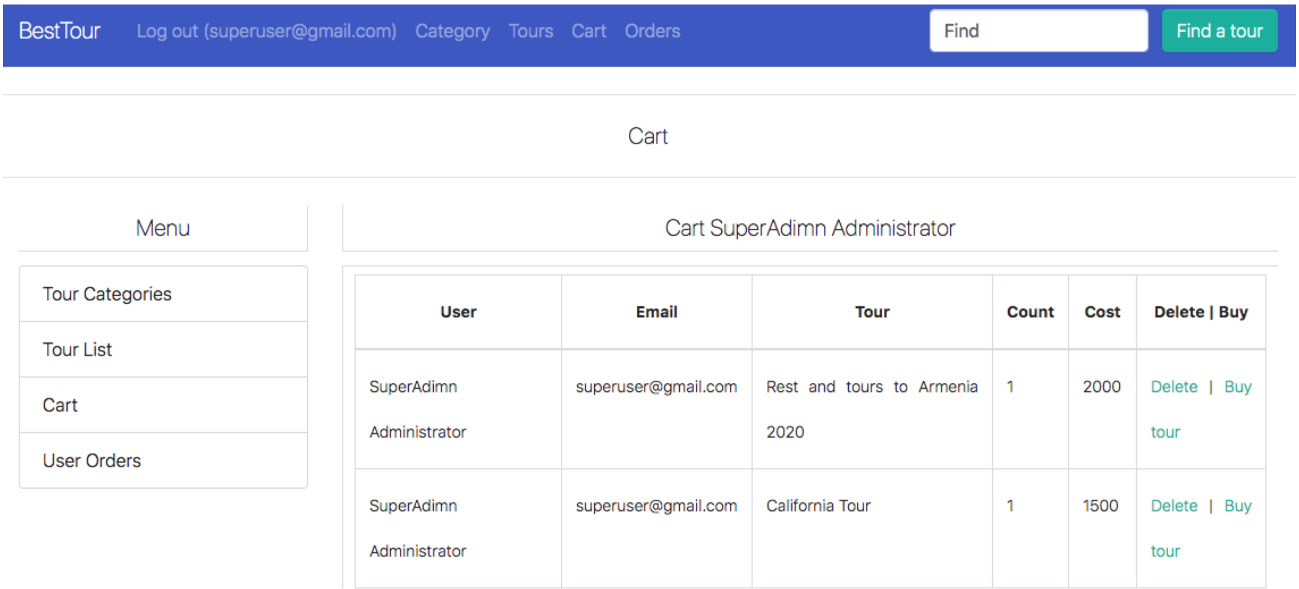
Category Description

Tours to Armenia

Picture 50 Category update page[74]

[73] Author’s desktop skreen

[74] Author’s desktop skreen

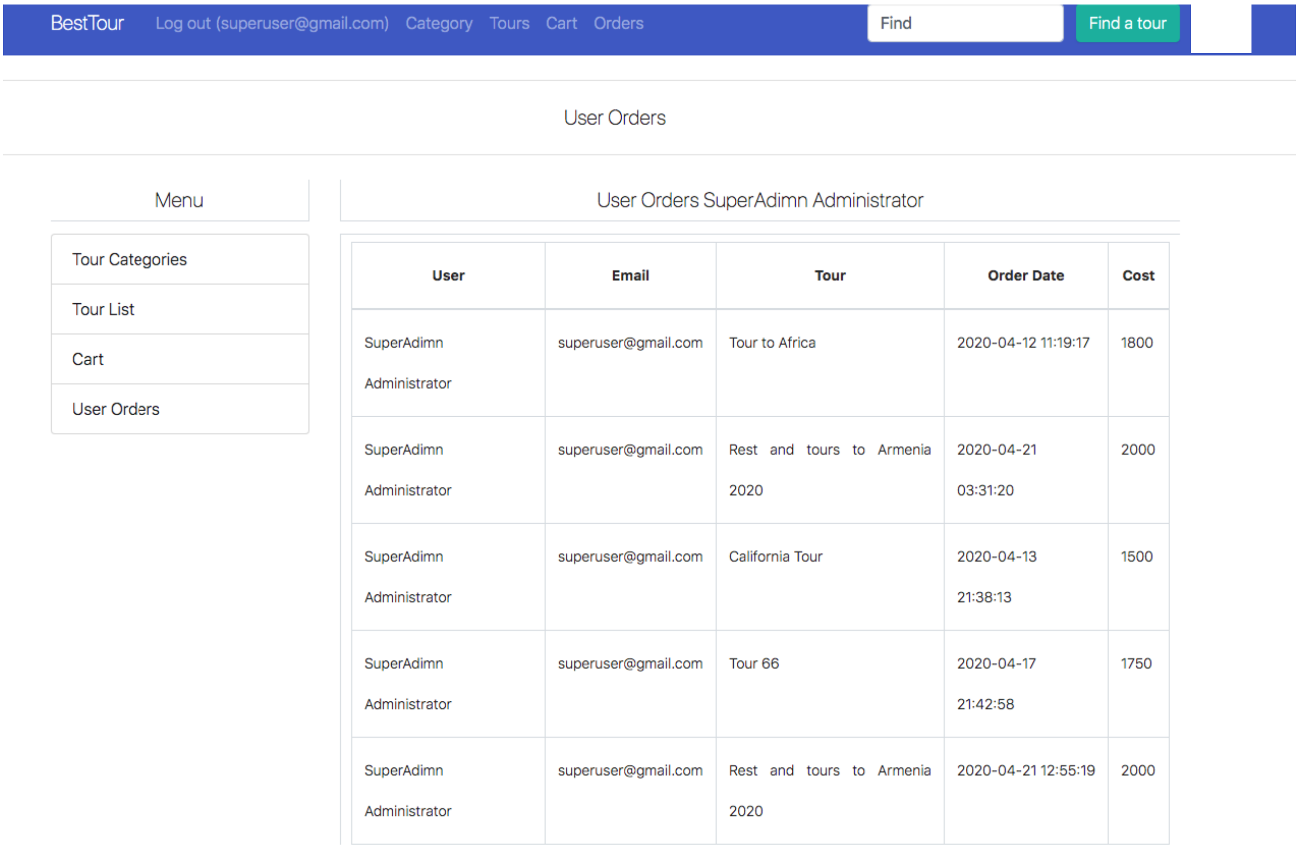


Cart

Cart SuperAdimn Administrator

User	Email	Tour	Count	Cost	Delete | Buy
SuperAdimn Administrator	superuser@gmail.com	Rest and tours to Armenia 2020	1	2000	Delete | Buy tour
SuperAdimn Administrator	superuser@gmail.com	California Tour	1	1500	Delete | Buy tour

Picture 51 User cart[75]



User Orders

User Orders SuperAdimn Administrator

User	Email	Tour	Order Date	Cost
SuperAdimn Administrator	superuser@gmail.com	Tour to Africa	2020-04-12 11:19:17	1800
SuperAdimn Administrator	superuser@gmail.com	Rest and tours to Armenia 2020	2020-04-21 03:31:20	2000
SuperAdimn Administrator	superuser@gmail.com	California Tour	2020-04-13 21:38:13	1500
SuperAdimn Administrator	superuser@gmail.com	Tour 66	2020-04-17 21:42:58	1750
SuperAdimn Administrator	superuser@gmail.com	Rest and tours to Armenia 2020	2020-04-21 12:55:19	2000

Picture 52 User orders[76]

[75] Author’s desktop skreen

[76] Author’s desktop skreen

Feature 2: SQL-script[77]

```
CREATE TABLE `tourcompany`.`role` (
 `idrole` INT(11) NOT NULL AUTO_INCREMENT,
 `role` VARCHAR(20) NULL DEFAULT NULL,
 PRIMARY KEY (`idrole`))
ENGINE = InnoDB
AUTO_INCREMENT = 4
DEFAULT CHARACTER SET = utf8;
CREATE TABLE `tourcompany`.`user` (
 `iduser` INT(11) NOT NULL AUTO_INCREMENT,
 `email` VARCHAR(40) NULL DEFAULT NULL,
 `username` VARCHAR(45) NULL DEFAULT NULL,
 `usersecondname` VARCHAR(45) NULL DEFAULT NULL,
 `password` VARCHAR(45) NULL DEFAULT NULL,
 `role_idrole` INT(11) NULL DEFAULT NULL,
 PRIMARY KEY (`iduser`),
 INDEX `fk_user_role1_idx` (`role_idrole` ASC),
 CONSTRAINT `fk_user_role1`
  FOREIGN KEY (`role_idrole`)
  REFERENCES `tourcompany`.`role` (`idrole`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = utf8;
CREATE TABLE `tourcompany`.`category` (
 `idcategory` INT NOT NULL AUTO_INCREMENT,
 `namecategory` VARCHAR(45) NULL,
 `description` MEDIUMTEXT NULL,
 PRIMARY KEY (`idcategory`))
ENGINE = InnoDB;
CREATE TABLE `tourcompany`.`tour` (
 `idtour` INT NOT NULL AUTO_INCREMENT,
 `name` VARCHAR(45) NULL,
 `description` MEDIUMTEXT NULL,
 `category_idcategory` INT NOT NULL,
 `cost` VARCHAR(45) NULL,
 PRIMARY KEY (`idtour`),
 INDEX `fk_tour_category1_idx` (`category_idcategory` ASC),
 CONSTRAINT `fk_tour_category1`
  FOREIGN KEY (`category_idcategory`)
  REFERENCES `tourcompany`.`category` (`idcategory`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE `tourcompany`.`cart` (
 `idcart` INT NOT NULL AUTO_INCREMENT,
 `user_iduser` INT(11) NOT NULL,
 `tour_idtour` INT NOT NULL,
```

[77] Made by author

```
 `count` INT NULL,
 PRIMARY KEY (`idcart`),
 INDEX `fk_cart_user1_idx` (`user_iduser` ASC),
 INDEX `fk_cart_tour1_idx` (`tour_idtour` ASC),
 CONSTRAINT `fk_cart_user1`
  FOREIGN KEY (`user_iduser`)
  REFERENCES `tourcompany`.`user` (`iduser`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
 CONSTRAINT `fk_cart_tour1`
  FOREIGN KEY (`tour_idtour`)
  REFERENCES `tourcompany`.`tour` (`idtour`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE `tourcompany`.`order` (
 `idorder` INT NOT NULL AUTO_INCREMENT,
 `date` DATE NULL,
 `cart_idcart` INT NOT NULL,
 PRIMARY KEY (`idorder`),
 INDEX `fk_order_cart1_idx` (`cart_idcart` ASC),
 CONSTRAINT `fk_order_cart1`
  FOREIGN KEY (`cart_idcart`)
  REFERENCES `tourcompany`.`cart` (`idcart`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```