**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ГАБРОВО**

Факултет: **Електротехника и електроника**

Катедра**: Автоматика, информационна и управляваща техника**

Р-л катедра: ..................................... Декан: ...................................................

/доц.. д-р инж. Д. Чантов/ /проф. д-р инж. Зв. Ненова/

**ДИПЛОМНА РАБОТА**

на студента: **Искрен Павлинов Иванов**

Name: **Iskren Pavlinov Ivanov**

ОКС **бакалавър** специалност **АИУТ** фак. № **21211106**

**Тема:**

**Уеб-базирана информационна система за организиране на почивки**

Консултант: ....................................... Ръководител: ...............................................

Студент: .......................................... /ас. инж. Фатме Рашидова/

/ **Искрен Иванов** /

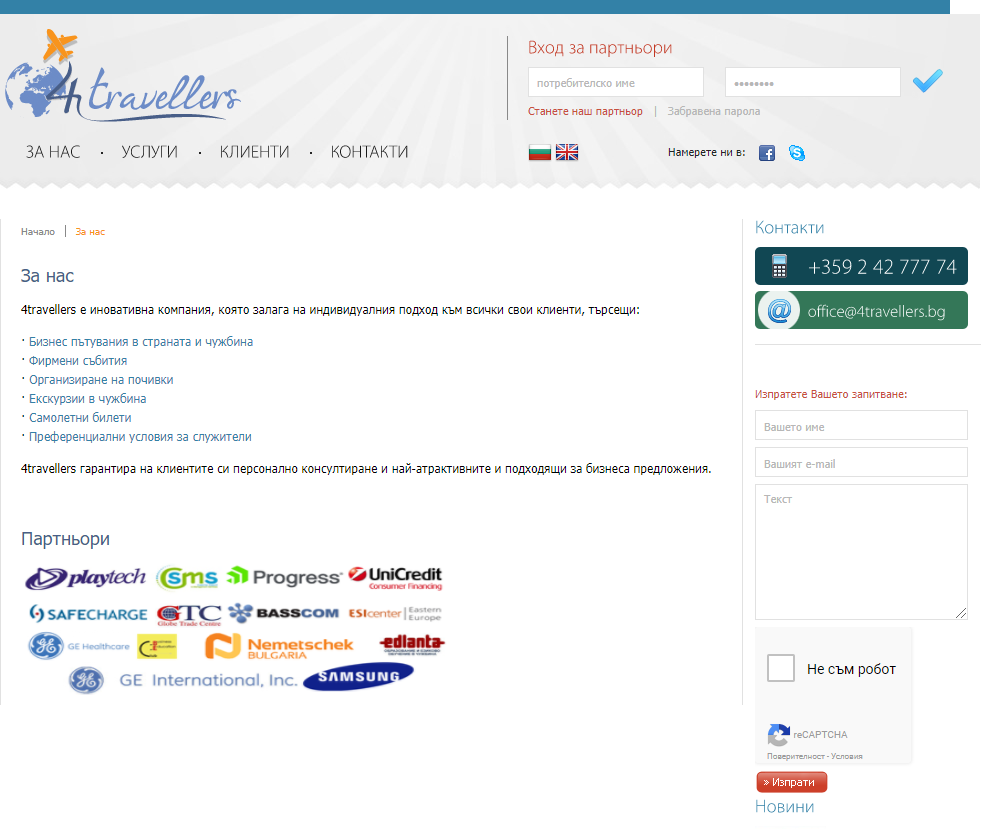
**Глава 1:**

**Обзор, изследване и анализиране на съществуващи Уеб-базирани системи за организиране на почивки. Предмет, цели и задачи на дипломната работа.**

**1.1 Обзор, изследване и анализиране на съществуващи Уеб-базирани системи за организиране на почивки.**

В интернет пространството е пълно със сайтове свързани с почивки и екскурзий, ще дам за пример 3 уебсайта(уеб-системи), които разполагат с огромна функционалност и същевременно е трудно да се ориентираш кое къде се намира. Моята система опростява това като налага точно 3 основни групи към които клиента да се насочи и да поръча и улеснява процеса по закупуване на почивка възможно най – много като прилага ясен потребителски интерфейс и бърза и динамична работа.

Обзор 1: http://4travellers.bg:



Плюсове:

- Изчистен дизайн

- Възможност за регистрация.

- Възможност за смяна на език.

- Възможност за споделяне на системата в социалните мрежи.

- Лесно достъпно поле за споделяне на мнение относно сайта.

Минуси:

- Сайтът не е оптимизиран за мобилни устройства.

- Объркващи менюта с неясна цел.

- Малък шрифт.

- Твърде много реклами.

- Незареждащи менюта.

- Няма имейл за потвърждаване на регистрация след регистрация.

Обзор 2: https://www.bohemia.bg****

Плюсове:

- Изчистен дизайн

- Възможност за регистрация.

- Търсачка

- Възможност за намиране на най-добра оферта за самолетен билет.

- Възможност за намиране на хотели на най-добри цени.

Минуси:

- Сайтът не е оптимизиран за мобилни устройства.

- Объркващи менюта с неясна цел.

- Малък шрифт.

- Твърде много реклами.

- Незареждащи менюта.

- Неработещи форми

- Няма имейл за потвърждаване на регистрация след регистрация.

- Проблеми с разплащателна система.

- Бавно зареждащи страници.

- Незареждащи иконки в различните браузъри.

Обзор 3: https://izgodni.bg/

Плюсове:

- Голям избор от услуги

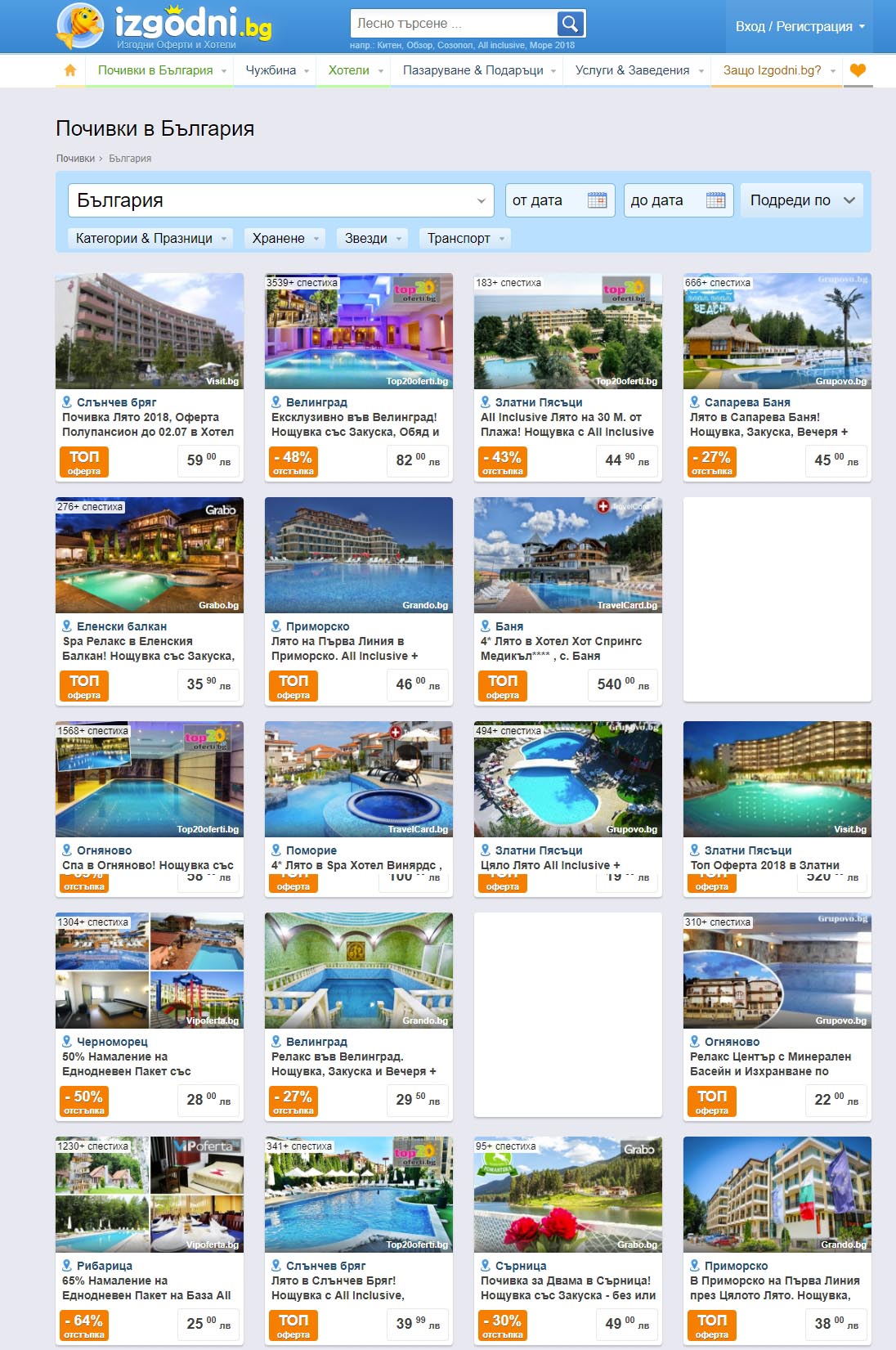
- Възможност за регистрация.

- Търсачка

- Възможност за намиране на завдения и магазини в изграната дестинация.

- Множество опции за пазаруване.

- Предлагане на процентни намаления.



Минуси:

- Сайтът не е оптимизиран за мобилни устройства.

- Объркващи менюта с неясна цел.

- Претрупан дизайн

- Неработеща търсачка

- Твърде много реклами.

- Незареждащи менюта.

- Неработещи форми

- Бавно зареждащи страници.

- Незареждащи иконки в различните браузъри.

**1.2 Предмет, цели и задачи на дипломната работа.**

**1.2.1 Предмет на дипломната работа:**

Тази дипломна работа е с цел изготвяне на уеб-система с помощта на която клиентите да могат да закупуват и намират почивки в красиви дестинации. Съкращаване и улесняване на процеса по намиране и поръчване на избраната почивка и същевременно опция за оценяване на самата почивка така че бъдещите клиенти да придобият ясна представа какво им се предлага. Създаване на поле за търсене на оферта и полета за коментари и оценка на дадената оферта.

**1.2.2 Цели и задачи на дипломната работа:**

Целите на тази диплломна работа са:

* Изграждане на уеб-система с помощта на която да могат да се поръчват екскурзии.
* Опростяване и улесняване на процеса по намиране на почивки.
* Система с която да могат да се търсят почивки по ключова дума.
* Клиентски панел чрез който всеки клиент да има възможност да се регистрира, пише коментари и оценява избраната от него почивка.
* Изчистен дизайн и достъпност както от Десктоп така и от мобилни устройства.
* Записване на час и дата на последно актуализирана оферта или публикуван коментар.
* Опростен и разбираем потребителски интерфейс.

**Глава 2:**

**Използвани методи, средства и инструменти за разработка на Уеб-базирана система.**

# Особености на уеб-базираните информационни системи

Web информационна система, или уеб-базирана информационна система, е информационна система, която използва Интернет уеб технологии да доставя информация и услуги, до потребители или други информационни системи / приложения. Това е софтуерна система, чиято основна цел е да публикува и поддържа данни чрез използване на хипертекстови-базирани принципи.

Информационна уеб система обикновено се състои от един или повече уеб приложения, специфични функции, ориентирани към компоненти, заедно с информационни елементи и други не-уеб компоненти. Web браузър обикновено се използва като front-end, докато базата данни като back-end.

Поради бързото развитие на уеб технологиите и на нарастващия интерес на потребителите и разработчиците, понятието за даден уеб сайт се движи от набор от HTML страници към уеб-базирана информационна система (WIS). A WIS е информационна система, осигуряваща средства за достъп до комплексни данни и интерактивни услуги чрез Мрежата. E- бизнес приложения, CRM и приложения на веригата за доставки, са примери за WIS.

Въпреки че има прилики между традиционната и уеб-базирани информационни системи, съществуват и значителни различия. Уеб-

базирани информационни системи представляват различни предизвикателства към разработчика, който трябва да се справят с неизвестни по отношение на условия мрежи и потребители, на различни типове данни, както и други въпроси, като например управление на съдържанието, представянето и използваемост. Въпреки, че уеб-базирани информационни системи са платформено независими по отношение на информацията за доставка, което е една от причините, поради които те са толкова популярни, уеб разработчиците трябва да се справят с много различни мрежи и трябва да ги вземат под внимание по време на процеса на разработване на системата. Потребителите на традиционен информационна система често са служители, които работят в рамките на една организация или служба. В контраст, анализ на изискванията на потребителите за WIS е далеч по-голямо предизвикателство за разработчиците.

# Езици за реализиране на уеб приложения

Скриптовите езици бързо се превръщат в [програмни езици](http://tuj.asenevtsi.com/ST2008/ST019.htm) с общо предназначение, поради тяхната функционалност. Те се използват масово там, където времето на разработчиците е по-ценно от времето за изпълнение на програмата. С тях се пишат програми, предназначени за:

Вграждане в [Web](http://tuj.asenevtsi.com/IntT/IntT084.htm)-страници. Тези програми не се компилират, а се интерпретират от браузъра при зареждането на съответната страница на клиентския компютър. С помощта на стандартните средства на скрипт- езика и на обектите, дефинирани от [DOM](http://tuj.asenevtsi.com/IntT/IntT045.htm), програмистът може да получи достъп до всеки елемент от Web-страницата и динамично да промени неговото съдържание и стилово оформяне.

Реализиране на разнообразни динамични ефекти като показване и скриване на съдържание, придвижване и позициониране на обекти,

анимация и т.н. Съединяване в едно цяло различни пакети [приложни](http://tuj.asenevtsi.com/ST2008/ST014.htm) [програми](http://tuj.asenevtsi.com/ST2008/ST014.htm). Създаване на големите [Интернет](http://tuj.asenevtsi.com/IntT/IntT003.htm) приложения. Автоматизиране.

За да се разбере по-добре същността на скриптовите езици е необходимо да се изходи от второто им наименование – езици за сценарии.

Сценарий – последователност от операции, които потребителят може да извършва на [компютъра](http://tuj.asenevtsi.com/CA/CA003.htm). Сценариите обикновено се интерпретират, а не компилират.

Приложна програма сценарий (скрипт) — програма, която автоматизира някаква задача, която без сценария потребителя трябва да извърши ръчно. Тъй като сценариите се интерпретират от изходния код динамично при всяко изпълнение, те се изпълняват значително по-бавно от готови програми, транслирани в машинен код на етап на компилиране.

Динамично изпълними скриптовите езици четат инструкциите от файла на програмата на минимални функционални блокове и изпълняват тези блокове без да четат останалия код.

Предварително компилирани скриптовите езици – прочитат програмата в началото, компилират я цялата в машинен код или някакъв вътрешен формат и след това я изпълняват.

Предимства на скриптовите езици. Скриптовите езици притежават по-сложен инструментариум и поддържат по-прогресивни техники на програмиране. Това съкращава времето за разработка на програми и ги прави по ефективни. Скриптовите езици позволяват бързо да се откриват грешки в програмата и да се реализират доработки на кода без да се изчаква приключване на компилацията. Скриптовите езици са сравнително лесни за изучаване, дори и без съответната подготовка.

# HTML (HyperText Markup Language)

HTML (HyperText Markup Language) или език за маркиране на хипертекст е основният език за описание на уеб страници. HTML е

стандарт в Интернет, а правилата се определят от международния консорциум W3C. Текущата версия на стандарта е HTML5.

Описанието на документа става чрез специални елементи или маркери, които се състоят от етикети или тагове (HTML tags). HTML елементите са основната градивна единица на уеб страниците. Чрез тях се оформят отделните части от текста на една уеб страница, като заглавия, цитати, раздели, хипертекстови препратки и т.н. Най-често HTML елементите са групирани по двойки и имат отварящ таг <h1> и затварящ таг </h1>.

HTML е основната технология, контролираща това, което уеб браузърът показва на екрана. Основните категории команди езика HTML, включени във версиите 3.2 и 4.01 са предназначени за: специфициране на стилови формати и управление на текстовия поток (Flow Control); включване на графични изображения (Images); създаване на хииервръзки (Links); интегриране на аудио с външни графични обекти (Sound and Maps); вмъкване на видео таг (Video) и на youtube клипове директно от тяхната страница, чрез тага (Iframe); създаване на интерактивни формуляри (Forms); разделяне на документа на отделни полета (Frames); включване на външни приложения (Applet), написани на езика Java; осъществяване на връзка с външни информационни структури (CGI-script). HTML е език за маркиране, който уеб браузърите използват, за да интерпретират и създават текст, изображения и други материали. Неизпълнената характеристика за всяка точка се определя в браузъра, като могат да бъдат променяни или засилвани с допълнителни дизайнерски

страници, наречени Cascading Style Sheets (CSS).

В повечето случаи HTML кодът е написан в текстови файлове и се хоства на сървъри, свързани към Интернет. Тези файлове съдържат текстово съдържание с маркери – инструкции за браузъра за това как да се показва текстът. Предназначението на уеб браузърите е да могат да

прочетат HTML документите и да ги превърнат в уеб страници. Браузърите не показват HTML таговете, а ги използват, за да интерпретират съдържанието на страницата.

Основното предимство на HTML е, че документите, оформени по този начин, могат да се разглеждат на различни устройства, а не само на екрана. Документът може да бъде правилно оформен и върху монитора на персонален компютър.

Създаването на HTML-базирана уеб страница може да се извърши с помощта на обикновен текстов редактор. Този начин изисква познаване на HTML тагове, така че те да бъдат интегрирани в текста, който ще се показва на страницата. Също така, често срещани са по-приятелски настроените инструменти, които не изискват от потребителя да притежава познания по HTML. Специализирани инструменти за създаване на HTML страници са някои от продуктите на Microsoft, FrontPage, Macromedia Dreamweaver, Notepad, Notepad++, Sublime Text, NetBeans и други.

Пример 1. Обикновен HTML код

<!DOCTYPE html>

<html>

<head>

<title>Заглавие, което се визуализира в таба на браузъра</title>

</head>

<body>

<h1>Заглавие</h1>

<p>Параграф, в който може да се публива

статия</p>

</body>

<html>

Семантични елементи в HTML5:

* Елемент <header> - представлява заглавие на елемент или група от елементи. Тага <header> позволява използването на няколко заглавия в един документ.
* Елемент <nav> - дефинира навигацията на сайта. Той е предназначен за големи блокове от навигационни връзки. Не всички връзки трябва да са в <nav> елемента, а само главното навигационно меню.
* Елемент <article> - представлява предмет/статия. Съдържанието в

<article> трябва да има смисъл само по себе си и да може да се разпространява независимо и отделно от останалата част на сайта. Използва се най-често за публикация във форум; публикация в блог; новини; коментари и др.

* Елемент <section> е тематично групирано съдържание, обикновено със заглавие. Този елемент позволява влагане на нова сегмантична структура във вече съществуващ елемент.
* Елемент <aside> - представя съдържание което е странично от основното съдържание на страницата. Съдържанието в <aside> се счита за неважно и може да бъде пропуснато, ако сайта се възпроизвежда на устройство с малък екран като телефон.
* Елемент <footer> - може да съдържа информация за автора на страницата; навигация на сайта; авторските права върху използваните материали и връзки към други (подобни) публикации.

# CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) е език за описание на стилове, който се използва основно за описване на представянето на документ, написан на език за маркиране (HTML, XML). Официално спецификацията на CSS се поддържа от W3C (World Wide Web Consortium). CSS позволява да се определя как да изглеждат елементите на една HTML страница –

шрифтове, размери, цветове, фонове и др. CSS кодът се състои от последователност от стилови правила, всяко от които представлява селектор, последван от свойства и стойности. CSS е удобен начин за форматиране на текст, шрифтове, списъци, таблици и др.

Всяко CSS правило има две основни части: селектор и една или повече декларации. Ако трябва да сме още по-прецизни, всяка декларация се състои от комбинацията свойство и стойност. Синтаксисът е следния:



*Фиг.1.1 CSS правило*

От лявата страна на свойството се намира селекторът – това е частта от правилото, указваща за кой елемент от документа трябва да се приложат стиловете. В примера са избрани всичките елементи на body. От дясната страна на свойство се намира декларацията, която представлява комбинация от свойството и стойност. Свойството е стилов атрибут, на който се задава стойност. Всяко свойство има стойност. CSS декларацията винаги завършва с точка и запетая, а групите от декларации са заградени от фигурни скоби.

Най-голямото улеснение, което предлага CSS е свързано с контрола на голям набор HTML документи, като контролирането на външния вид на страниците може да става по няколко начина:

* + - * Външни стилове (External Style Sheet), когато е създаден отделен файл с разширение .css, който съдържа само стилове. Тогава не е нужно да се променя HTML кода във всяка една от HTML страниците.

Пример 1. CSS код с добавени стилове в </head> частта.

<h1>Заглавие</h1>

<p >Параграф, в който може да се публива

статия</p>

</body>

<html>

href="external.css"

rel="stylesheet"

<link type="text/css" />

</head>

<body>

content="text/html; charset=UTF-8" />

Content-Type

http-equiv=

<meta

<!DOCTYPE html>

<html>

<head>

<title>CSS като отделен файл </title>

Пример 2. CSS код с добавени стилове в </head> частта.

<!DOCTYPE html>

<html>

<head>

<title> вграден CSS файл </title>

<meta http-equiv= Content-Type content="text/html;

charset=UTF-8" />

<style type=”text/css”>

</style>

</head>

<body>

<h1>Заглавие</h1>

<p >Параграф, в който може да се публива статия</p>

</body>

<html>

* + - * Вградени в самите HTML елементи.

</style>

</head>

<body>

<h1>Заглавие</h1>

<p style=”text-align: left;”>Параграф, в който може да се публива статия</p>

</body>

<html>

content="text/html; charset=UTF-8" />

<style type=”text/css”>

Content-Type

http-equiv=

<meta

<!DOCTYPE html>

<html>

<head>

<title> вграден CSS файл </title>

Пример 3. CSS код със стилове в самите HTML елементи.

При наличие на повече от един стил, специфициран за HTML елемент, ще се използва този, който има най-висок приоритет (четири е с най-високият), както следва:

1. Стилът на браузъра по подразбиране.
2. Външните стилове.
3. Вътрешните стилове.
4. Стиловете, вградените в самите HTML елементи.

Първоначално HTML етикетите били създадени, за да определят съдържанието на документа. Или иначе казано идеята била следната: таговете трябвало да “казват”: “Това е заглавие”, “Това е параграф”, “Това е таблица”, използвайки синтаксис от рода <h1> (за големи заглавия), <p> (за параграф). Постепенно с добавянето на нови спецификации към HTML и с увеличаване на изискванията към визията на сайта, ставало все по- трудно да се създават страници само с него.

Друго предимство на CSS е, че спестява работа и по-точно писане на код. Тъй като можем във външен файл да съхраним повтарящите се форматирания и така да не пишем всеки път атрибутите за тях. Освен това, по този начин при евентуална промяна на изгледа на едно форматиране, се налага само промяна в съответния стил. Например, ако имаме 10 страници с шрифт Arial, размер 2, цвят черен, и искаме да променим цвета, но не сме задали стил. То ще се наложи ръчно да го направим във всичките страници поотделно. Задавайки стил за конкретното форматира в отделен файл, трябва само да поправите там.

Синтаксисът на CSS е съставен от три части: селектор (selector), характеристика (property) и стойност (value).

selector {property: value}

Селекторът е нормален HTML елемент (таг), който трябва да се промени, характеристиката е атрибутът за смяна, като всяка характеристика си има стойност. Характеристиката и стойността са отделени с двоеточие и са заградени с фигурни скоби:

body {color: black}

Ако стойността представлява низ съдържащ няколко думи – необходимо е да се огради с кавички:

p {font-family: "sans serif"}

Ако трябва да се специфицира повече от една характеристика, е необходимо да се разделят с “;”.

p {text-align:center;color:red}

Селекторите с еднакви характеристики могат да се групират по между си и така да се спести писане на код. Това става като се разделят със запетая. В следващия пример са групирани всички header елементи. След прилагането на стила съответния текст ще бъде оцветен в зелено:

h1,h2,h3,h4,h5,h6 { color: green }

CSS притежава още едно важно свойство, така нареченият клас Селектор (class selector). С него може да се зададат различни стилове за един и същи HTML елемент в уеб страницата. Ако например имаме два параграфа, единият от тях е с дясно подравняване а другият с централно, с клас селектор това би изглеждало така:

p.right {text-align: right}

p.center {text-align: center}

Друга полезна опция в CSS e id селекторът. Той се различава от класа селектор, по това че последния може да се приложи върху няколко елемента върху страницата, а id селекторът е приложим само за един. Той трябва да е уникален в рамките на уеб страницата. Долу е дефиниран стил, който свързва елемента <p> с id стойността “paragraph1”

p# paragraph1{text-align: center; color: red }

Добра практика е като се пише код да се оставят коментари, такава възможност притежава и CSS. Коментарът започва с "/\*" и завършва с "\*/". Всичко между тях се игнорира от браузъра.

# JavaScript

# JavaScript (чете се *джаваскрипт*) е интерпретируем [език за програмиране](https://bg.wikipedia.org/wiki/%D0%95%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5), разпространяван с повечето [Уеб браузъри](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D1%8A%D1%80). Поддържа [обектно-ориентиран](https://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%B0%D0%BD) и [функционален](https://bg.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) стил на програмиране. Създаден е в [Netscape](https://bg.wikipedia.org/w/index.php?title=Netscape&action=edit&redlink=1) през 1995 г. Най-често се прилага към [HTML](https://bg.wikipedia.org/wiki/HTML)-а на [Интернет страница](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82_%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) с цел добавяне на функционалност и [зареждане на данни](https://bg.wikipedia.org/wiki/AJAX). Може да се ползва също за писане на сървърни скриптове [JSON](https://bg.wikipedia.org/wiki/JSON), както и за много други приложения. JavaScript не трябва да се бърка с [Java](https://bg.wikipedia.org/wiki/Java), съвпадението на имената е резултат от маркетингово решение на Netscape. Jav[S](https://bg.wikipedia.org/wiki/Java)script е стандартизиран под името [Java](https://bg.wikipedia.org/wiki/Java).

### Зареждане

JavaScript кодът може да се вмъква в HTML документа между двойката елементи <script> и</script>. Когато срещне тага <script>, браузърът разбира, че трябва да спре интерпретирането на [HTML](https://bg.wikipedia.org/wiki/HTML) кода и да започне да обработва скрипта, намиращ се между <script> и </script>. Този скрипт не е задължително да бъде написан на JavaScript. Има и други езици за писане на скриптове, например [VBScript](https://bg.wikipedia.org/w/index.php?title=VBScript&action=edit&redlink=1). Но езикът по подразбиране е JavaScript. В следния пример създаваме уеб страница в която е вмъкнат JavaScript код, който да изписва на екрана „Здравей!“

1 <!Doctype HTML>

2 <html>

3 <head>

4 <title>javascript</title>

5 </head>

6 <body>

7 <script>

8 document.write("Здравей!");

9 </script>

10 </body>

11 </html>

Препишете горния код в някой текстов редактор и запазете файла като hi.html. След това го отворете с някой браузър.

Другият начин да заредите JavaScript е да го поставите в отделен файл. В следващия пример отново ще напишем „Здравей“ на екрана, но ще заредим скрипта си от отделен файл.

Първо създаваме файла Hello.js:

document.write("Здравей!");

След което трябва да променим Hello.html от предходния пример по следния начин:

1 *<!--*

2 *<script>*

3  *document.write("Здравей!");*

4 *</script>*

5 *-->*

6

7 <script src="Hello.js"></script>

Това разделение на javascript логиката от HTML структурата е един от основните принципи на [Софтуерното инженерство](https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80%D0%BD%D0%BE_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D1%81%D1%82%D0%B2%D0%BE).

### Променливи[[редактиране](https://bg.wikipedia.org/w/index.php?title=JavaScript&veaction=edit&section=7) | [редактиране на кода](https://bg.wikipedia.org/w/index.php?title=JavaScript&action=edit&section=7)]

Променливите в JavaScript са динамични и могат да съдържат стойности от всеки тип. Декларират се чрез ключовата дума var следвана от името на променливата или списък от променливи. Възможно е също да присвоите стойност на променливата в момента на декларация чрез оператора =. По подразбиране стойността на променливата ще бъде undefined.

var greeting = "Здравей!", x = 42;

document.write(greeting);

Тук имаме списък от променливи. Името на първата е greeting, а стойността ѝ е „Здравей!“. Kогато отпечатваме променливата greeting ние всъщност отпечатваме стойността ѝ „Здравей!“.

#### Имена на променливи[[редактиране](https://bg.wikipedia.org/w/index.php?title=JavaScript&veaction=edit&section=8) | [редактиране на кода](https://bg.wikipedia.org/w/index.php?title=JavaScript&action=edit&section=8)]

Имената на променливите в JavaScript не трябва да започват с цифра, могат да започват с малка или голяма буква, \_ или $ и да съдържат само тези знаци както и числа. Всички останали символи са забранени. Името на променливата не може да е ключова дума или оператор.

*//Валидни имена на променливи:*

var ninja, apples\_and\_oranges, var56, $, nutsAndBolts;

*//Невалидни имена на променливи:*

var 3, var, "test", if, while, 5wtx1;

#### Обхват на променливите[[редактиране](https://bg.wikipedia.org/w/index.php?title=JavaScript&veaction=edit&section=9) | [редактиране на кода](https://bg.wikipedia.org/w/index.php?title=JavaScript&action=edit&section=9)]

В JavaScript има два вида обхват за променливите: функционален и глобален.

* Функционален обхват важи за всяка променлива обявена чрез var в рамките на дадена функция. Тази променлива е видима единствено в рамките на тази функция.
* Глобален обхват важи за всяка променлива обявена извън функция или без var, дори и във функция. Тази променлива е видима от всеки друг код на страницата.

#### Константи[[редактиране](https://bg.wikipedia.org/w/index.php?title=JavaScript&veaction=edit&section=10) | [редактиране на кода](https://bg.wikipedia.org/w/index.php?title=JavaScript&action=edit&section=10)]

В JavaScript няма константи. В по-новите версии на езика даден обект може да се замрази object.freeze(myConstants); което спира промяната на стойностите на членовете му.

### Прости типове данни[[редактиране](https://bg.wikipedia.org/w/index.php?title=JavaScript&veaction=edit&section=11) | [редактиране на кода](https://bg.wikipedia.org/w/index.php?title=JavaScript&action=edit&section=11)]

В JavaScript има няколко основни прости типа данни:

* Числа – всички числа в JavaScript са представени по един и същи начин в паметта. Няма отделна категории за цялочислени и дробни.
* Низове – низовете се ограждат с '' или "" *и могат да съдържат всеки друг знак.*
* Булеви – само две стойности: true и false.

*//Инициализиране на числова променлива:*

var a = 3;

*//Инициализиране на променлива с низ*

var b = "JavaScript iz kuwl";

*//Инициализиране на булева променлива*

var c = true;

*//Извеждане на типа на данните в променливите*

alert(typeof a);

alert(typeof b);

alert(typeof c);

Този пример създава променливи с начална стойност от три различни типа данни, поддържани от JavaScript. Първо използваме ключовата дума typeof, за да върне типовете данни и след това ги подаваме като параметри на функцията alert за да бъдат показани в диалогов прозорец.

# 1.2.4 SQL

SQL или Език за структурирани запитвания (на [английски](https://bg.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8_%D0%B5%D0%B7%D0%B8%D0%BA): *Structured Query Language, SQL*) е популярен [език за програмиране](https://bg.wikipedia.org/wiki/%D0%95%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5), предназначен за създаване, видоизменяне, извличане и обработване на [данни](https://bg.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D0%B8) от [релационни системи за управление на бази данни](https://bg.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B8_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8_%D0%B7%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0_%D0%B1%D0%B0%D0%B7%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8). Стандартизиран е от [ANSI](https://bg.wikipedia.org/wiki/ANSI) / [ISO](https://bg.wikipedia.org/wiki/ISO).

*SQL* обикновено се произнася *ес кю ел*(*ess-cue-el*) (виж [английската азбука](https://bg.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B0_%D0%B0%D0%B7%D0%B1%D1%83%D0%BA%D0%B0)) — или *сикуъл* (*sequel*). Имената на продукти, съдържащи *SQL*, имат всеки свое произношение, например: [MySQL](https://bg.wikipedia.org/wiki/MySQL)официално се произнася *май ес кю ел*(*my ess cue el*); [PostgreSQL](https://bg.wikipedia.org/wiki/PostgreSQL) - *постгрес*(*postgres*); и [Microsoft SQL Server](https://bg.wikipedia.org/wiki/Microsoft_SQL_Server) на български като *Майкрософт ес кю ел сървър*, а иначе като *Майкрософт сикуъл сървър* (*Microsoft-sequel-server*).

Освен стандартния SQL, дефиниран от ANSI и ISO, съществуват още много негови разширения и вариации. На практика почти всички разработчици предоставят различни вариации и разширения на SQL, които в литературата често се описват като SQL *диалекти*. Много от тези разширения са със затворен характер, например Oracle [PL/SQL](https://bg.wikipedia.org/w/index.php?title=PL_SQL&action=edit&redlink=1), [IBM](https://bg.wikipedia.org/wiki/IBM) SQL PL (SQL Procedural Language) и [Sybase](https://bg.wikipedia.org/w/index.php?title=Sybase&action=edit&redlink=1) / [Microsoft](https://bg.wikipedia.org/wiki/Microsoft) [Transact-SQL](https://bg.wikipedia.org/w/index.php?title=Transact-SQL&action=edit&redlink=1). Общото за повечето комерсиални изпълнения е да пропускат поддръжката на основни възможности, заложени в стандарта, като DATE или TIME типове данни, предпочитайки техен собствен вариант. В резултат SQL кодът много рядко може да бъде "пренесен" между различни архитектури без да се модифицира, и то значително, за разлика от ANSI [C](https://bg.wikipedia.org/wiki/C_(%D0%B5%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)) или ANSI [Fortran](https://bg.wikipedia.org/wiki/Fortran), които могат да се "пренасят" без големи промени. Има няколко причини за това:

* сложността и размерът на SQL стандарта означават, че повечето бази от данни не осъществяват стриктно целия стандарт.
* стандарта не специфицира как базата данни се държи в няколко важни области (като индексите(indexes)), оставяйки на изпълняващия го да реши по какъв начин да става то.
* SQL стандартът специфицира синтаксиса, на който базата данни трябва да отговаря. Но спецификацията на семантиката на езика е по-слабо дефинирана и позволява различно тълкуване.
* много доставчици на бази от данни имат множество [клиенти](https://bg.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D1%8A%D1%80%D0%B2%D1%8A%D1%80) и ако SQL стандартът е в конфликт с начина на работа на предишни инсталирани версии на тяхната база от данни, те не искат да я променят заради обратната съвместимост.

SQL е създаден със специфична, ограничена цел — запитвания към данни в релационна база от данни. Като такъв той е [множествено-базиран](https://bg.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE), [декларативен](https://bg.wikipedia.org/w/index.php?title=%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%B5%D0%BD_%D0%B5%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5&action=edit&redlink=1)[език за програмиране](https://bg.wikipedia.org/wiki/%D0%95%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5) а не [процедурен език](https://bg.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5&action=edit&redlink=1) като [C](https://bg.wikipedia.org/wiki/C_(%D0%B5%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)) или [BASIC](https://bg.wikipedia.org/wiki/BASIC) които, бидейки [неспециализирани](https://bg.wikipedia.org/wiki/%D0%95%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5), са направени да разрешават по-голям обхват от проблеми. Разширения на езика като [PL/SQL](https://bg.wikipedia.org/w/index.php?title=PL_SQL&action=edit&redlink=1) намаляват разликата добавяйки процедурни елементи, и контролни структури. Друг подход е да се позволи вграждане на кода. Например Oracle и други включват [Java](https://bg.wikipedia.org/wiki/Java) в базата данни, докато [PostgreSQL](https://bg.wikipedia.org/wiki/PostgreSQL)позволява функциите за бъдат написани на различни езици, като [Perl](https://bg.wikipedia.org/wiki/Perl), [Tcl](https://bg.wikipedia.org/wiki/Tcl), и [C](https://bg.wikipedia.org/wiki/C_(%D0%B5%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)).

Популярна шега относно SQL е че "SQL не е структуриран, нито ограничен до запитвания, нито език (на [английски](https://bg.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8_%D0%B5%D0%B7%D0%B8%D0%BA): *is not structured, nor is it limited to queries, nor is it a language*)." Основа за това е, че *чистият* SQL не е класически програмен език, тъй като не отговаря на *Дефиницията на Тюринг*. От друга страна обаче той е език за програмиране, защото има граматика, синтаксис и е предназначен за програмиране. Тази шега наподобява забележката на [Волтер](https://bg.wikipedia.org/wiki/%D0%92%D0%BE%D0%BB%D1%82%D0%B5%D1%80) относно [Свещената римска империя](https://bg.wikipedia.org/wiki/%D0%A1%D0%B2%D0%B5%D1%89%D0%B5%D0%BD%D0%B0%D1%82%D0%B0_%D1%80%D0%B8%D0%BC%D1%81%D0%BA%D0%B0_%D0%B8%D0%BC%D0%BF%D0%B5%D1%80%D0%B8%D1%8F), че не е била "нито свещена, нито Римска, нито империя" /"not holy, nor Roman, nor an empire."/

SQL е в контраст с по-мощните [езици за програмиране от четвърта генерация](https://bg.wikipedia.org/w/index.php?title=%D0%95%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5_%D0%BE%D1%82_%D1%87%D0%B5%D1%82%D0%B2%D1%8A%D1%80%D1%82%D0%B0_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F&action=edit&redlink=1), ориентирани към бази от данни като [Focus](https://bg.wikipedia.org/w/index.php?title=Focus_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0&action=edit&redlink=1) или [SAS](https://bg.wikipedia.org/w/index.php?title=SAS_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&redlink=1) с неговата относителна функционална простота и по-прости команди. Това намалява трудностите по поддържането на [изходния код](https://bg.wikipedia.org/wiki/%D0%98%D0%B7%D1%85%D0%BE%D0%B4%D0%B5%D0%BD_%D0%BA%D0%BE%D0%B4) на SQL, но също прави по-трудно програмирането на въпроси като 'Кой имаше 10-те най-високи резултата?', което води до разработката на процедурни разширения. Обаче то прави възможно изходния SQL код да се създава (и оптимизира) от програма, водещо до разработката на "естествени" езици за запитвания към бази от данни, и '[довлечи и пусни](https://bg.wikipedia.org/w/index.php?title=%D0%94%D0%BE%D0%B2%D0%BB%D0%B5%D1%87%D0%B8_%D0%B8_%D0%BF%D1%83%D1%81%D0%BD%D0%B8&action=edit&redlink=1)' пакети за програмиране на бази от данни с '[обектно ориентирани](https://bg.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D0%B5%D0%BA%D1%82%D0%BD%D0%BE_%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%B0%D0%BD&action=edit&redlink=1)' интерфейси. Често те позволяват преглед на резултатния SQL изходен код за подобряване, с образователна цел или за да се използва в друга среда.

**1.3 Технологии за изработване на уеб-базираните информационни системи**

**1.3.1 Bootstrap**

Bootstrap, първоначално наречена Twitter Blueprint, е разработена от Марк Ото и Джейкъб Торнтън в Twitter като рамка за насърчаване на последователността във вътрешните инструменти. Преди Bootstrap бяха използвани различни библиотеки за разработка на интерфейса, което доведе до несъответствия и голямо натоварване за поддръжка. Според разработчик на Twitter Марк Ото:

"Една супер малка група от разработчици и се събрахме, за да проектираме и изградим нов вътрешен инструмент и видяхме възможност да направим нещо повече. Посредством този процес видяхме, че изграждаме нещо много по-съществено от друго вътрешно средство. с ранна версия на Bootstrap, като начин за документиране и споделяне на общи дизайнерски модели и активи в компанията. "

След няколко месеца на развитие от малка група много разработчици в Twitter започнаха да допринасят за проекта като част от Hack Week, седмица в стил hackathon за екипа за развитие на Twitter. Той бе преименуван от Twitter Blueprint до Bootstrap и пуснат като проект с отворен код на 19 август, 2011. Тя продължава да бъде поддържана от Марк Ото, Якоб Торнтън и малка група от основни разработчици, както и голяма общност от сътрудници.

На 31 януари 2012 г. бе пусната Bootstrap 2, която добави система за оформление на мрежата от 12 колони, вградена поддръжка за Glyphicons, няколко нови компонента, както и промени в много от съществуващите компоненти.

На 19 август 2013 г. бе пусната Bootstrap 3, която преработи компонентите за използване на плоския дизайн и мобилния първи подход.

На 29 октомври 2014 г. Марк Ото обяви, че Bootstrap 4 се развива. Първата алфа версия на Bootstrap 4 бе пусната на 19 август 2015 г. Първата бета версия бе пусната на 10 август 2017 г. Стабилната версия на Bootstrap 4 бе пусната на 18 януари 2018 г. основно подобрявайки стиловете на печат, помощните средства за границите и осигурява по-голям контрол върху гъвкавата кутия.

Характеристика

Bootstrap 3 поддържа най-новите версии на Google Chrome, Firefox, Internet Explorer, Opera и Safari (с изключение на Windows). Освен това поддържа обратно към IE8 и най-новата версия на разширената поддръжка на Firefox (ESR).

От 2.0, Bootstrap поддържа отзивчив уеб дизайн. Това означава, че оформлението на уеб страниците се настройва динамично, като се вземат предвид характеристиките на използваното устройство (настолен, таблетен, мобилен телефон).

Започвайки с версия 3.0, Bootstrap прие философия на дизайна за мобилни устройства, като по подразбиране акцентира върху дизайна, който отговаря на изискванията.

Версия 4.0 алфа издание добавя Sass и flexbox подкрепа.

**1.3.1 Node.JS**

**Node.js** е мулти-платформена среда за изпълнение на сървърни и мрежови приложения с отворен код. Приложенията се пишат на JavaScript и могат да се изпълняват в Node.js среда под OS X, Microsoft Windows, Linux и IBM.

Node.js предоставя задвижвана от събития архитектура и неблокираща входно-изходна система за програмиране на приложенията (API), която оптимизира производителността и мащaбируемостта на дадена програма. Тези технологии са често използвани за мрежови приложения, работещи в реално време.

Node.js използва JavaScript двигателя V8 на Google, за да изпълнява код и голяма част от основните модули са написани на JavaScript. Node.js съдържа вградени библиотеки, които позволяват приложенията да работят като сървъри, без софтуер като Apache HTTP Server или IIS.

Node.js позволява създаването на [уеб сървъри](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D1%81%D1%8A%D1%80%D0%B2%D1%8A%D1%80) и мрежови инструменти, използвайки [JavaScript](https://bg.wikipedia.org/wiki/JavaScript) и колекция от модули, които боравят с различните функционалности на ядрото.  Модулите се грижат за входно-изходната система, работата в мрежа ([HTTP](https://bg.wikipedia.org/wiki/HTTP), [TCP](https://bg.wikipedia.org/wiki/TCP), [UDP](https://bg.wikipedia.org/wiki/User_Datagram_Protocol), [DNS](https://bg.wikipedia.org/wiki/Domain_Name_System), или [TLS](https://bg.wikipedia.org/wiki/TLS) / [SSL](https://bg.wikipedia.org/wiki/SSL)), двоичните данни (буфери), [криптографските](https://bg.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) функции, потока от данни, както и други основни функции.  Модулите на Node.js имат семпъл и елегантен приложно-програмен интерфейс, което от своя страна намалява сложността при писането на сървърни приложения.

Могат да бъдат използвани софтуерни рамки, с цел да бъде ускорено разработването на приложения – достъпни за публично ползване са софтуерните рамки Express.js, Socket.io и Connect. Node.js приложения могат да работят под [Microsoft Windows](https://bg.wikipedia.org/wiki/Microsoft_Windows), [Unix](https://bg.wikipedia.org/wiki/Unix) и [Mac OS X](https://bg.wikipedia.org/wiki/OS_X) сървъри. Node.js приложения, като алтернатива, могат да бъдат написани с CoffeeScript (по-четима форма на JavaScript), Microsoft TypeScript (силно типизирана форма на JavaScript), или който и да е друг програмен език, на който може да се компилира JavaScript.

Node.js се използва предимно за изграждане на мрежови програми като уеб сървъри, което го прави подобен на [PHP](https://bg.wikipedia.org/wiki/PHP) и [Python](https://bg.wikipedia.org/wiki/Python).  Най-голямата разлика между PHP и Node.js е, че PHP е език за блокиране (командите се изпълняват само след като предишната команда е завършила с изпълнението си), докато при Node.js изпълнението протича по различен начин(командите се изпълняват паралелно и използват обратни извиквания, за да сигнализират своето завършване).

Node.js дава възможност за развитие на бързи [уеб сървъри](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D1%81%D1%8A%D1%80%D0%B2%D1%8A%D1%80) под JavaScript. [[23]](https://bg.wikipedia.org/wiki/Node.js#cite_note-b1-23)Разработчиците могат да създават високо мащабируеми сървъри. Това става с помощта на опростен модел, който използва обратни извиквания към сигнала за завършване на командата.  Node.js е създаден, защото е трудно постигането на съгласуваност при много сървърни езици за програмиране, което често води до лошо изпълнение. Node.js свързва лекотата на скриптов език (JavaScript) с мощността на Unix мрежово програмиране. 

Node.js е създадена на двигателя на [Google](https://bg.wikipedia.org/wiki/%D0%93%D1%83%D0%B3%D1%8A%D0%BB) V8 JavaScript, защото:

* V8 е с отворен код под лиценза на [BSD](https://bg.wikipedia.org/wiki/BSD)
* V8 е изключително бърз
* V8 е фокусиран върху уеб, така че може да се каже че е специалист в областта на не малко основи в интернет пространството като [HTTP](https://bg.wikipedia.org/wiki/HTTP), [DNS](https://bg.wikipedia.org/wiki/Domain_Name_System), [TCP](https://bg.wikipedia.org/wiki/TCP)
* [JavaScript](https://bg.wikipedia.org/wiki/JavaScript) е добре познат език в рамките на [уеб програмирането](https://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5), което го прави достъпен за много уеб разработчици

Хиляди библиотеки с [отворен код](https://bg.wikipedia.org/wiki/%D0%9E%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD_%D0%BA%D0%BE%D0%B4) са били създадени за Node.js и могат да бъдат изтеглени безплатно от интернет страницата на проекта. Node.js разчита на общност от разработчици, концентрирани около два адресни списъка и [IRC](https://bg.wikipedia.org/wiki/Internet_Relay_Chat) канал # node.js на freenode. Обществото от програмисти, занимавали се с Node.js се събира на [NodeConf](http://nodeconf.com/), годишна конференция фокусирана върху Node.js. [[](https://bg.wikipedia.org/wiki/Node.js#cite_note-30)

## **Технически характеристик**

### Threading

Node.js работи върху една нишка, като не блокира входно-изходната комуникация, което му позволява да поддържа десетки хиляда връзки едновременно, без да се налага да се притесняваме за забавяне при контекстното превключване между нишките. Дизайнът на споделяне на една нишка между всички искания означава, че може да бъде използван за изграждане на високо паралелни приложения. Целта на дизайна на Node.js приложението е при всяка входящо-изходяща операция да се използва обратно повикване.

Недостатък на този подход е, че Node.js не позволява мащабиране в зависимост от ядрата на процесора на клиента, на който бива изпълняван, ако не се използват допълнителни модули като pm2.

### V8

V8 е двигателят за JavaScript, създаден специално за браузъра Google Chrome, чийто код Google отваря през 2008 година. Написан под С++, V8 компилира изходния код (JIT) до машинен код на момента, замествайки интерпретацията в реално време.

Node.js съдържа libuv, който пък от своя страна поема асинхронните събития. V8 осигурява средата за изпълнение за JavaScript. Libuv е слой за абстракция на мрежата, както и за функционалността на файловата система под Windows и POSIX-базирани системи като Linux, Mac OS X и Unix.

Главната функционалност на Node.js е внедрена в JavaScript библиотеката. Свързващите части на Node.js, писани на С++ свързват всички тези технологии една с друга, както и с операционната система.

### Управление на пакети

npm е предварително инсталиран пакетен мениджър (опция за управление на пакетите) за сървърна платформа от типа Node.js. Този мениджър се използва за инсталацията на Node.js програмите от npm регистъра. Организирайки инсталирането и управлението на външни Node.js програми, той позволява на програмистите да изграждат приложенията си по-бързо. npm не трябва да се бърка с CommonJS require() командата, защото той не се използва за зареждане на код, а за инсталиране на код и управление на неговите зависимости от командния ред. Пакетите, които се намират в npm регистъра варират от прости спомагателни библиотеки като underscore.js до по-сложни мениджъри на задачи като Grunt.

### Унифицирано АПИ

В комбинация с браузер, база данни за документи (от типа на MongoDB или CouchDB), както и с JSON, Node.js предоставя унифициран JavaScript стек за разработване. Със засиленото внимание спрямо клиентските рамки, както и с адаптирането на основно сървърни шаблони за разработка като MVC, MVP, MVVM и така нататък, Node.js позволява преизползването на същия модел и обслужващ интерфейс, както за сървъра така и за клиентите.

### Цикъл на събитията

Node.js се регистрира с операционната система, за да бъде уведомяван когато бъде направена връзка. Когато се направи връзка, операционната система провежда обратно повикване. В рънтайма на Node.js всяка връзка е малка алокация в динамичната памет. Традиционно всяка връзка се поема от сравнително тежки системни процеси или нишки. Node.js използва цикъл на събитията вместо процеси или нишки, за да поеме милиони връзки едновременно. За разлика от други сървъри базирани на събития, цикъла на събитията на Node.js няма нужда да се вика изрично. Вместо това се дефинират обратните повиквания и сървъра автоматично влиза в цикъла на събитията в края на дефиницията. Node.js излиза от цикъла, когато няма вече обратни повиквания за изпълнение.

# Предимства на уеб-базираните информационни системи

Най-важните предимства на уеб-базираните системи на XML от последно поколение могат да се резюмират както следва:

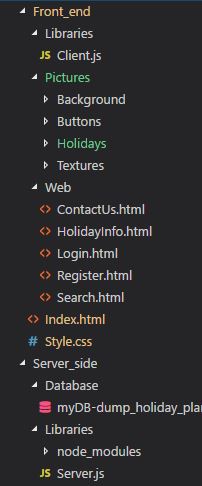
* 24-часа / 7 дни в седмицата;
* Диверсификацията на достъпа. Можете да определите нивото за достъпа до системата на всеки от служителите си;
* Достъп от всяко място – няма инсталация на определен компютър. С помощта на интернет браузър и персонална парола входа в системата можете да стане от всеки свързан с интернет персонален компютър, лаптоп и дори смартфон. По този начин служителите ви винаги ще имат достъп до системата, независимо къде се намират;
* Единна база данни. Централизираните уеб системи позволяват съхранението на всички данни на едно място. По този начин отпада консолидирането на данните от различните персонални компютри и офиси  
  на туроператора. Не се налага да се инсталира отделна база данни във всеки компютър, в който се използва системата;
* Инсталацията, настройката и поддръжката на софтуера е уеб базирана. Вие не се нуждаете тези действия да се извършават на всеки компютър поотделно. Всички промени и модификации по системата се извършват само на едно място (сървъра на системата), без да е необходимо да се отразяват по различните работни места. Системата е една и се ползва от всички потребители чрез сървър. Това прави поддръжката й ефективна, бърза и евтина;
* Възможността за интеграцията на системата позволява базата данни от автори, издателства, заглавия и преводачи да може да се интегрира и на вашата интернет страница;
* Не се налага ръчно да експортирате данни.

**Глава 3.**

**Архитектура на системата. Проектиране на базата от данни на системата. Логическа и физическа релационна схема. Специфични моменти при реализиране на системата.**

**3.1 Архитектура на системата.**

**3.1.1 Файлова структора:**



Уеб системата е изградена от няколко основни компонента:

Front End: Съвкупност от файлове изграждащи потребителският интерфайс.

Web: файлове писани на HTML изграждащи основните страници на системата като основна е index.html или началната страница която сървъра отваря при зареждане. Всяка HTML страница определя определени части от системата.

**Index.html:**

Изграден от няколко основни таблици отговарящи за основните компоненти на страницата. Осномите елементи са “div” и “table”.

**ContactUs.html:**

Отговаря за информацията на фирмата или бизнеса

**Holidayinfo.html:**

Страницата, която се зарежда при отваряне на определена почивка, като е изградена от секция за сменящи се снимки, поле за вот от клиента, описание на почивката и коментари от страна на клиентите.

**LogIn.html:**

Страницата отговаряща за регистрация от клиентите, които вече са си направили профил в системата. Тази страница е изградена от два основни форма: Име и Парола, като паролата е със скрити символи с цел конфиденциалност на клинета.

**Register.html:**

Страницата отговаря за създаване на профил на клинета в систематая изградена е от форм полета, които събират основните данни от потребителя нужни за създаване на профил и ги изпращат към базата данни.

**Search.html:**

Страницата е предназначена за търсачка на оферти, разполага с форма за търсене и бутон.

**Style.css**

В този файл се записват всички CSS стилове които стилизират HTML страниците. Изграден е от множество класове и ИД-та задаващи начина на стилизиране на html елементите.

В папка Libraries се съдържа функционалността на клиентската част от уеб-системата, основния файл вътре е Client.js, който съдържа JavaScript код отговарящ за динамиката на системата.

Съдържа няколко основни функции, които изграждат клиентската страна на системата. Register() функцията отговаря за взимането на данните от формата за регистрация и е съвкупност от правила с цел коректно задаване на данните от потребителя. XMLHttpRequest обекти за взаимодействие със сървърите. Можете да изтегляте данни от URL адрес, без да е необходимо да обновявате цялата страница. Това позволява на уеб страницата да актуализира само част от страницата, без да нарушава това, което потребителят прави. XMLHttpRequest се използва много в програмирането с AJAX.

Libraries: папката отговаряща за клиентската логика на системата, съдържаща един JavaScript код, който отговаря за динамиките по html страниците и функционалностите, които те притежават, като събиране на данни от потребителя и т.н. Pictures: съвкупност от папки съдържащи всички изображения под формата на .JPG формат, които се използват в системата.

Server Side e папката, която съдържа всички скриптове и файлове отговарящи за базите с данни и основните компоненти,които сървърът приема. Като технoлогии за създаването на сървърната логика са използвани Node.js, SQL,MySQL Workbench и MySQL Server.В папката Database се намира така наречения дъмп файл, писан на SQL, използван за съхранение на информацията от уеб страниците в базата данни.Изграждан е чрез MySQL Workbench и за стартирането на системата е необходимо да бъде зареден в сървъра. Изграден е чрез множество от таблици определящи начина на зареждане на всички данни в базата с данни. Следващата папка е Libraries: съдържа два основни компонента

* Server.js

Кодът, който отговаря за сървърната част на системата. Използва три основни компонента за връзка със сървъра и обмен на данни:

HTTP модул: модулът може да създаде HTTP сървър, който слуша сървърните портове и дава отговор на клиента.

URL модул: URL адресният модул разделя уеб адреса на четими части

MySQL модул: този модул се използва за манипулиране на базите данни на MySQL

Останалата част от кода е съвкупност от функционалности, определящи какво ще се запише в базите при въвеждането на информация от потребители

* node\_modules:

Съвкупност от JavaScript библиотеки отговарящи за функционалноста на модулите в Server.js файла.

**3.2.Проектиране на базата от данни на системата**

Базата от данни е изградена от 6 таблици всяка със своята спецификация. Таблицата roles или роли, отговаря за правата на клиента, Admin или Member. Admin правата дават възможност на потребителя да трие и променя данни в системата. Member правата дават възможост единствено за харесване или коментиране на дадена екскурзия.

Таблица Users е таблицата в която се записват всички потребители регистрирали се в системата изградена е от 10 колони като всяка отговаря за дадена информация на клиента. Връзката между Roles и Users e едно към много.

Следващата основна таблица е holidays отговаряща за данните на екскурзиите които се въвеждат от администратора. Основната им специфика е че си взаимодействат със всички останали таблици.Разполага с колони информация, снимки, цена и т.н.

Таблица votes е таблицата която съхранява вота на клиента изградена от 4 основни колони ID(първичен ключ), HolidayID отговаряща за данните взети от Holiday таблицата. Votevalue вот от клинета и nickname – име на клиента.Връзката между holidays и votes е „много към много“.

Таблица comments e таблицата в която се съхраняват коментарите от потребителите и има връзка с holidays и user „много към много“.

Таблица orders записва данните при поръчка от клиента като прави връзка с users и comments.

**3.3 Логическа и физическа релационна схема**

**3.3.1 Логическа релационна схема**

Roles:

-ID(primary key)

-Edit Site

-EditComments

-VoteHoliday

Users:

-Nickname(primary key)

- Password

- Role

- Email

- First Name

- Sure Name

- Last Name

- Country

- Town

- Adress

Comments:

-ID(primary key)

-Comment

-Nickname

-HolydayID

-DateTime

Holydays:

-ID(primary key)

-Holyday Name

-Destinations

-Days

-Price

-Start Date

-End Date

-Pictures URL

-Short Description

-Description

-VoteUpCount

-VoteDowncount

Votes:

-ID(primary key)

-HolidayID

-Votevalue

-Nickname

Orders:

-ID(primary key)

-Description

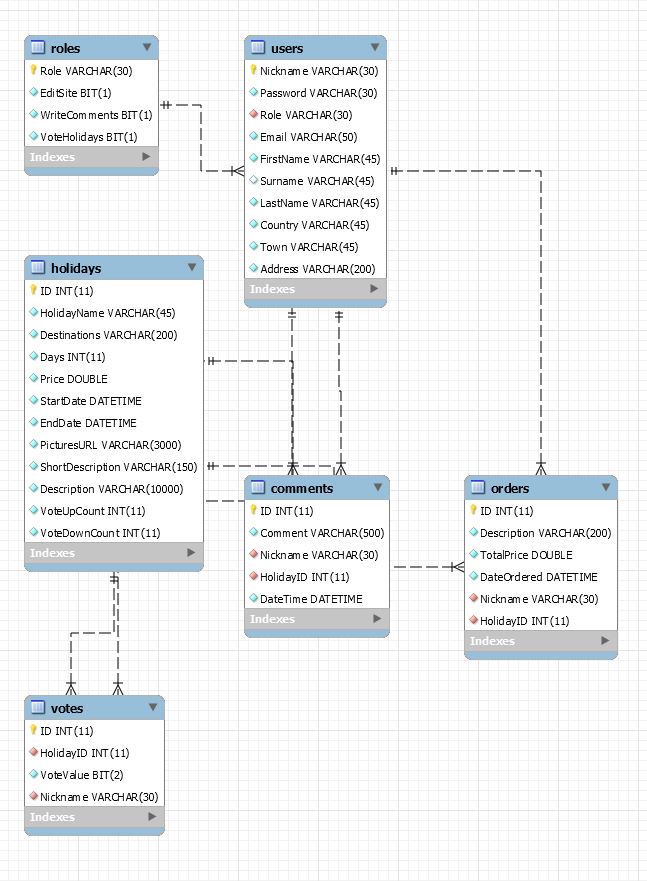
-Total Price

-Date Ordered

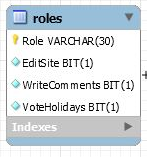
-Nickname

-HolidayID

**3.3.2 Физическа релационна схема**



**Описание на таблици:**

****

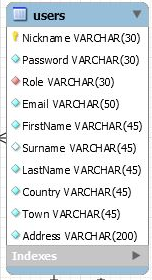
**Roles(**Роли**):**

Role VARCHAR(30) – първичен ключ с дължина до 30 символа

EditSite BIT(1) – поле за проверка тип бит и дължина 1 символ

WriteComments BIT(1) – поле за разрешаване на коментари тип бит и дължина 1 символ

VoteHolidays BIT(1) - поле за разрешаване на вот тип бит и дължина 1 символ



**Users**(Потребители)**:**

Nickname VARCHAR(30) – име на потребителя,първичен ключ с дължина до 30 символа тип varchar

Password VARCHAR(30) – поле за парола на потребителя с дължина до 30 символа и тип VARCHAR

Role VARCHAR(30) – поле за роли тип VARCHAR и дължина 30 символа

Email VARCHAR(50) – поле за имейл тип VARCHAR и дължина 50 символа

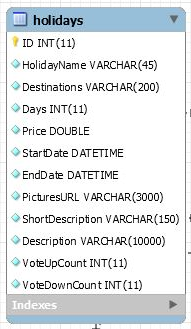
FirstName VARCHAR(45) – поле за въвеждане на име на потребителя тип VARCHAR и дължина 45 символа

LastName VARCHAR(45) - поле за въвеждане на име на потребителя тип VARCHAR и дължина 45 символа

Country VARCHAR(45) - поле за въвеждане на държав на потребителя тип VARCHAR и дължина 45 символа

Town VARCHAR(45) - поле за въвеждане на град на потребителя, тип VARCHAR и дължина 45 символа

Address VARCHAR(200) - поле за въвеждане на адрес на потребителя, тип VARCHAR и дължина 200 символа



**Holidays(**Екскурзии**):**

ID INT(11) – първичен ключ, номер на дадената почивка тип INT и дължина до 11 символа

HolidayName VARCHAR(45) – поле за име на екскурзията тип VARCHAR и дължина до 45 символа

Destination VARCHAR(200) – поле за описание на мястото на екскурзията тип VARCHAR и дължина до 200 символа

Days INT(11) – поле за въвеждане на дните на екскурзията тип INT и дължина до 11 символа

Price DOUBLE – поле за въвеждане на цена на екскурзията тип DOUBLE

StartDate DATETIME – поле за въвеждане на начална дата на екскурзията тип DATETIME

EndDate DATETIME - поле за въвеждане на крайна дата на екскурзията тип DATETIME

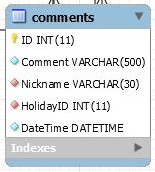
PicturesURL VARCHAR(3000) – поле за въвеждане на линк за снимка на екскурзията тип VARCHAR и дължина до 3000 символа

ShortDescription VARCHAR(150) – поле за кратко описание на екскурзията с дължина до 150 символа и тип VARCHAR

Description VARCHAR(10000) – поле за описание на екскурзията с дължина до 10000 символа и тип VARCHAR

VoteUpCount INT(11) – поле за въвеждане на положителен вот на екскурзията тип INT и дължина до 11 символа

VoteDownCount INT(11) – поле за въвеждане на отрицателен вот на екскурзията тип INT и дължина до 11 символа



**Comments(**Коментари**)**:

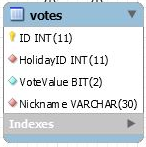
ID INT(11) – първичен ключ, номер на коментар тип INT и дължина до 11 символа

Comment VARCHAR(500) – коментар тип VARCHAR и дължина до 500 символа

Nickname VARCHAR(30) – име на потребителя тип VARCHAR и дължина до 30 символа

HolidayID INT(11) – ID номер на почивката тип INT и дължина до 11 символа

DateTime DATETIME – дата на коментар тип DATETIME



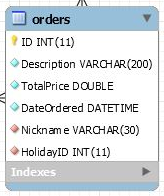
**Votes**(Вот):

ID INT(11) – първичен ключ, номер на вот, тип INT и дължина до 11 символа

HolidayID INT(11) – ID номер на почивката тип INT и дължина до 11 символа

VoteValue BIT(2) – стойност на вот, тип BIT и дължина до 2 символа

Nickname VARCHAR(30) – име на потребителя тип VARCHAR и дължина до 30 символа



**Orders(**Поръчки**):**

ID INT(11) – първичен ключ, номер на поръчка, тип INT и дължина до 11 символа

Description VARCHAR(200) – Описание на поръчка, тип VARCHAR и дължина до 200 символа

TotalPrice DOUBLE – Максимална цена на поръчка, тип DOUBLE

DateOrdered DATETIME – дата на поръчката, тип DATETIME

Nickname VARCHAR(30) - име на потребителя тип VARCHAR и дължина до 30 символа

HolidayID INT(11) – ID номер на почивката тип INT и дължина до 11 символа.

**3.4 Специфични моменти при реализиране на системата.**

За реализирането на системата е необходимо да имате конфигуриран локален MySQL Server, който можете да изтеглите от тук: <https://dev.mysql.com/doc/refman/8.0/en/> , за инсталация просто следвайте стъпките през MySQL installer.

Като за целта ще ви е необходимо да имате МySQL Installer, можете да изтеглите от тук:

https://dev.mysql.com/downloads/installer/

След инсталиране и конфигуриране на MySQL Server ще ви е необходим МySQL Workbench, чрез който да управляване и променяте sql файла в който се намират всичите таблици с данни на системата. Като можете да го изтеглите от тук: <https://www.mysql.com/products/workbench/>

Също може да използвате МySQL Installer за целта.

След като сте инсталирали всичко необходимо нужно за сървъра и базите данни е необходино да имате инсталиран и актуализиран Node.js, който можете да изтеглите от тук: <https://nodejs.org/en/>

**Стартиране на системата:**

След като вече имате всичко необходимо за работа със системата, стартирането става чрез зареждането на Server.js, който се намира в папка Server\_side/Libraries/. Използвайте командата node Server.js за да стартирате файла.

Следващата стъпка е зареждането на дъмп файла в МySQL Workbench, като за целта е необходино да създадете нова връзка, като за UserName използвайте „root“, Port:3306, HostName: 127.0.0.1

След създаването на връзка отидете на главното меню Server > Data Import > Import From Disk. Сложете тикче на Import From Self-Contained File и изберете дъмп файла на системата, който се намира в Server\_side/Database с име: myDB-dump\_holiday\_planner.sql

На следващото поле Default Target Schema изберете New и изберете „mysql1“ за име на схема. След това цъкнете Start Import и в SCHEMAS ще видите базата с данни на системата.

След като вече имате всичко готово за локално стартиране на системата отидете на index.html намиращ се в папка Front\_End и го отворете с Mozzila Firefox за най-оптимално и бързо зареждане.

**Глава 4:**

**Ръководство за работа със системата**

Още при отварянето на Index.htm файла ще видите всички екскурзии на първата страница, натискайки бутона „Home“ всеки път ще бъдете препращани към нея. За да отворите всяка една от екскурзиите просто трябва да кликнете на нея. Имате опции за смяна на снимките чрез които опции да резглеждате снимките на екскурзията кликайки на стрлките под основната снимка.

След това ще видите че можете да оставите вот на всяка от екскурзиите било то положителен в зелено или отрицателен в червено. За да оставите вот трябва да имате регистрация в системата. Като след това ще видите най-актуалната цена и описание на екскурзията. Имате възможност да четете или пишете коментари на всяка една от екскурзиите стига да имате профил, като просто натиснете Comment бутона в най-десния ъгъл на страницат.

В навигационния бар ще видите 3 полета „Home”, “Search” и „Contact US”

Home бутона пренасочва винаги към началната страница.

Search бутона ви пренасочва към търсачка на екскурзии, която намира по ключова дума екскурзията коя търсите. Примерно ако напишете Самоков ще излезе екскурзията в което описание присъства ключовата дума „Самоков“.

ContacUs бутона ще ви пренасочи към страницата с контакти на компанията която предлага екскурзии, в нея ще видите точната локация на компанията в Google maps и номер и име на фирмата под локацията.

В полето до навигационния бар ще видите и полето за регистрация. Като имате опциите за регистрация и вход в акаунта.

Register бутона ще ви пренасочи към полетата за регистрация на потребител, където имате 8 задължителни полета, които трябва да попълните ако желаете да се регистрирате в системата.

Login бутона ще ви пренасочи към полетата за вход на потребители.

Регистрираните потребители могат да оставят коментари, дават оцента или директно да поръчат дадената екскурзия.Като за целта е нужно да отворите екскурзията от която се интересувата и ще видите под снимките опция Buy Now, тази опция позволява записването на дадената екскурзия в базата с данни откъдето администратор да се свърже с вас за да потвърди поръчката. Друг основен компонент за регистрираните потребители е оставянето на коментар под описанието на екскурзията като натиснат бутона Comment.

За администратори или хората които искат да обновяват и променят почивките в системата ще е нужно да имат MySQL Workbench чрез, който да могат да променят необходимите данни.

Като за публикуване на нова екскурзия ще ви е необходимо да отворите таблица Holidays и от там да актуализирате, триете или добавяте нови екскурзии.

Ако искате да променяте правата на потребителите в системата трябва да отворите Users таблицата от където имате 2 възможности за всеки потребител, Admin и Member, като ролята Admin дава на потребителя правото да прави промени по сайта, докато Member ролята позволява единствено писане на коментари и оцента на екскурзията.

**Глава 5:**

**Приложение на първичен код на системата**

**5.1 Сорс код**

**5.1.1 Клиентска част:**

**HTML:**

**-------------------------------------------------------------------------------------------------**

Index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="Style.css">

<script type="text/javascript" src="Libraries/Client.js">

</script>

<title>

Holidays

</title>

</head>

<body onload="Index\_OnLoad()">

<div class="page">

<div class="mainTopPanel">

<div class="mainTopPanelContainer">

<span class="mainTopPanelButton">

<a href="Index.html">

<img src="Pictures/Buttons/ButtonHome.png" alt="Home" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonHome\_OnMouseEnter(this);" onmouseleave="ButtonHome\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="Web/Search.html">

<img src="Pictures/Buttons/ButtonSearch.png" alt="Search" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonSearch\_OnMouseEnter(this);" onmouseleave="ButtonSearch\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="Web/ContactUs.html">

<img src="Pictures/Buttons/ButtonContactUs.png" alt="Contact Us" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonContactUs\_OnMouseEnter(this);" onmouseleave="ButtonContactUs\_OnMouseLeave(this);">

</a>

</span>

</span>

</span>

</div>

</div>

<div class="pageContent">

<table style="width: 100%;

border: 0px;

text-align: center;">

<!-- ------------------------------------------------------------------------------- -->

<tr>

<td style="width: 49.5%">

<table id="tableHoliday1">

<tr>

<td>

<img id="imgHoliday1" alt="imgHoliday1" style="width: 100%;

height: auto;

border: 0;

visibility: hidden;">

</td>

</tr>

<tr>

<td>

<span id="priceHoliday1" class="priceStyle" style="visibility: hidden;">

</span>

</td>

</tr>

<tr>

<td>

<span id="shortDescriptionHoliday1" class="shortDescriptionStyle" style="visibility: hidden;">

</span>

</td>

</tr>

</table>

</td>

<td>

<span style="visibility: hidden">

Space

</span>

</td>

<td style="width: 49.5%">

<table id="tableHoliday2">

<tr>

<td>

<img id="imgHoliday2" alt="imgHoliday2" style="width: 100%;

height: auto;

border: 0;

visibility: hidden;">

</td>

</tr>

<tr>

<td>

<span id="priceHoliday2" class="priceStyle" style="visibility: hidden;">

</span>

</td>

</tr>

<tr>

<td>

<span id="shortDescriptionHoliday2" class="shortDescriptionStyle" style="visibility: hidden;">

</span>

</td>

</tr>

</table>

</td>

</tr>

<!-- ------------------------------------------------------------------------------- -->

<tr>

<td>

<span style="visibility: hidden">

Space

</span>

</td>

</tr>

<!-- ------------------------------------------------------------------------------- -->

<td style="width: 49.5%">

<table id="tableHoliday3">

<tr>

<td>

<img id="imgHoliday3" alt="imgHoliday3" style="width: 100%;

height: auto;

border: 0;

visibility: hidden;">

</td>

</tr>

<tr>

<td>

<span id="priceHoliday3" class="priceStyle" style="visibility: hidden;">

</span>

</td>

</tr>

<tr>

<td>

<span id="shortDescriptionHoliday3" class="shortDescriptionStyle" style="visibility: hidden;">

</span>

</td>

</tr>

</table>

</td>

<td>

<span style="visibility: hidden">

Space

</span>

</td>

<td style="width: 49.5%">

<table id="tableHoliday4">

<tr>

<td>

<img id="imgHoliday4" alt="imgHoliday4" style="width: 100%;

height: auto;

border: 0;

visibility: hidden;">

</td>

</tr>

<tr>

<td>

<span id="priceHoliday4" class="priceStyle" style="visibility: hidden;">

</span>

</td>

</tr>

<tr>

<td>

<span id="shortDescriptionHoliday4" class="shortDescriptionStyle" style="visibility: hidden;">

</span>

</td>

</tr>

</table>

</td>

</tr>

<!-- ------------------------------------------------------------------------------- -->

<tr>

<td>

<span style="visibility: hidden">

Space

</span>

</td>

</tr>

<!-- ------------------------------------------------------------------------------- -->

<tr>

<td style="width: 49.5%">

<table id="tableHoliday5">

<tr>

<td>

<img id="imgHoliday5" alt="imgHoliday5" style="width: 100%;

height: auto;

border: 0;

visibility: hidden;">

</td>

</tr>

<tr>

<td>

<span id="priceHoliday5" class="priceStyle" style="visibility: hidden;">

</span>

</td>

</tr>

<tr>

<td>

<span id="shortDescriptionHoliday5" class="shortDescriptionStyle" style="visibility: hidden;">

</span>

</td>

</tr>

</table>

</td>

<td>

<span style="visibility: hidden">

Space

</span>

</td>

<td style="width: 49.5%">

<table id="tableHoliday6">

<tr>

<td>

<img id="imgHoliday6" alt="imgHoliday6" style="width: 100%;

height: auto;

border: 0;

visibility: hidden;">

</td>

</tr>

<tr>

<td>

<span id="priceHoliday6" class="priceStyle" style="visibility: hidden;">

</span>

</td>

</tr>

<tr>

<td>

<span id="shortDescriptionHoliday6" class="shortDescriptionStyle" style="visibility: hidden;">

</span>

</td>

</tr>

</table>

</td>

</tr>

<!-- ------------------------------------------------------------------------------- -->

</table>

<!-- ------------------------------------------------------------------------------- -->

<table style="padding-left: 42%; padding-top: 5%; border: 0px;">

<tr>

<td>

<img id="imgPreviousPage" alt="imgPreviousPage" src="Pictures/Buttons/ButtonArrowLeft-Grey.png" style="width: auto;

height: auto;

border: 0;">

</td>

<td>

<span id="spanCurrentPage">

</span>

</td>

<td>

<img id="imgNextPage" alt="imgNextPage" src="Pictures/Buttons/ButtonArrowRight-Grey.png" style="width: auto;

height: auto;

border: 0;">

</td>

</tr>

</table>

<!-- ------------------------------------------------------------------------------- -->

</div>

</div>

<div class="rightBanner">

<div class="rightBannerMiddleLeft">

<a href="Web/Register.html" id="buttonRegister">

<img src="Pictures/Buttons/ButtonRegister.png" alt="Register" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonRegister\_OnMouseEnter(this);" onmouseleave="ButtonRegister\_OnMouseLeave(this);">

</a>

<span id="spanUserNames" style="display: none;">

User Names

</span>

</div>

<div class="rightBannerMiddleRight">

<a href="Web/Login.html" id="buttonLogin">

<img src="Pictures/Buttons/ButtonLogIn.png" alt="LogIn" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogin\_OnMouseEnter(this);" onmouseleave="ButtonLogin\_OnMouseLeave(this);">

</a>

<a href="Index.html" id="buttonLogout" onclick="Logout()" style="display: none">

<img src="Pictures/Buttons/ButtonLogOut.png" alt="LogOut" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogout\_OnMouseEnter(this);" onmouseleave="ButtonLogout\_OnMouseLeave(this);">

</a>

</div>

</div>

</body>

</html>

ContactUS.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../Style.css">

<script type="text/javascript" src="../Libraries/Client.js">

</script>

<title>

Holidays - Contact us

</title>

</head>

<body onload="ContactUs\_OnLoad()">

<div class="page">

<div class="mainTopPanel">

<div class="mainTopPanelContainer">

<span class="mainTopPanelButton">

<a href="../Index.html">

<img src="../Pictures/Buttons/ButtonHome.png" alt="Home" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonHome\_OnMouseEnter(this);" onmouseleave="ButtonHome\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="Search.html">

<img src="../Pictures/Buttons/ButtonSearch.png" alt="Search" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonSearch\_OnMouseEnter(this);" onmouseleave="ButtonSearch\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="ContactUs.html">

<img src="../Pictures/Buttons/ButtonContactUs.png" alt="Contact Us" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonContactUs\_OnMouseEnter(this);" onmouseleave="ButtonContactUs\_OnMouseLeave(this);">

</a>

</span>

</span>

</span>

</div>

</div>

<div class="pageContent">

<h1>

Contact us:

</h1>

<table>

<tr>

<td style="width: 45%;">

RelaxHolidays АД

</td>

<td style="width: 10%">

&nbsp&nbsp&nbsp

</td>

<td style="width: 45%;">

RelaxHolidays Ltd.

</td>

</tr>

<tr>

<td style="width: 45%;">

София, ул. "Хаджи Димитър" №26

</td>

<td style="width: 10%">

&nbsp&nbsp&nbsp

</td>

<td style="width: 45%;">

Sofia, st. "Hadji Dimitar" 26

</td>

</tr>

<tr>

<td style="width: 45%;">

GSM: +359 87 888 7750

</td>

<td style="width: 10%">

&nbsp&nbsp&nbsp

</td>

<td style="width: 45%;">

GSM: +359 87 888 7750

</td>

</tr>

<tr>

<td style="width: 45%;">

Искрен Иванов

</td>

<td style="width: 10%">

&nbsp&nbsp&nbsp

</td>

<td style="width: 45%;">

Iskren Ivanov

</td>

</tr>

</table>

</div>

</div>

<div class="rightBanner">

<div class="rightBannerMiddleLeft">

<a href="Register.html" id="buttonRegister">

<img src="../Pictures/Buttons/ButtonRegister.png" alt="Register" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonRegister\_OnMouseEnter(this);" onmouseleave="ButtonRegister\_OnMouseLeave(this);">

</a>

<span id="spanUserNames" style="display: none">

User Names

</span>

</div>

<div class="rightBannerMiddleRight">

<a href="Login.html" id="buttonLogin">

<img src="../Pictures/Buttons/ButtonLogIn.png" alt="LogIn" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogin\_OnMouseEnter(this);" onmouseleave="ButtonLogin\_OnMouseLeave(this);">

</a>

<a href="ContactUs.html" id="buttonLogout" onclick="Logout()" style="display: none">

<img src="../Pictures/Buttons/ButtonLogOut.png" alt="LogOut" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogout\_OnMouseEnter(this);" onmouseleave="ButtonLogout\_OnMouseLeave(this);">

</a>

</div>

</div>

</body>

</html>

HolidayInfo.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../Style.css">

<script type="text/javascript" src="../Libraries/Client.js">

</script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>

Holidays - Info

</title>

</head>

<body onload="HolidayInfo\_OnLoad();" onresize="HolidayInfo\_OnResize();">

<div class="page">

<div class="mainTopPanel">

<div class="mainTopPanelContainer">

<span class="mainTopPanelButton">

<a href="../Index.html">

<img class="rounded" src="../Pictures/Buttons/ButtonHome.png" alt="Home" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonHome\_OnMouseEnter(this);" onmouseleave="ButtonHome\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="Search.html">

<img class="rounded" src="../Pictures/Buttons/ButtonSearch.png" alt="Search" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonSearch\_OnMouseEnter(this);" onmouseleave="ButtonSearch\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="ContactUs.html">

<img class="rounded" src="../Pictures/Buttons/ButtonContactUs.png" alt="Contact Us" onmouseenter="ButtonContactUs\_OnMouseEnter(this);" onmouseleave="ButtonContactUs\_OnMouseLeave(this);">

</a>

</span>

</span>

</span>

</div>

</div>

<div class="pageContent">

<table style="width: 100%;

border: 0px;">

<tr>

<td style="height: 100px;

width: auto;

text-align: center;" colspan="3">

<img class="rounded" id="imgHoliday" alt="imgHoliday" style="width: auto;

height: 90%;

border: 0;">

</td>

</tr>

<tr>

<td style="padding-left: 40%;" colspan="3">

<span>

<img class="rounded" id="imgPreviousImage" alt="imgPreviousImage" src="../Pictures/Buttons/ButtonArrowLeft-Grey.png" style="width: auto;

height: auto;

border: 0;"> &nbsp;

<span id="spanCurrentImage">

</span>

&nbsp;

<img class="rounded" id="imgNextImage" alt="imgNextImage" src="../Pictures/Buttons/ButtonArrowRight-Grey.png" style="width: auto;

height: auto;

border: 0;">

</span>

</td>

</tr>

<tr>

<td style="width: 100%;

text-align: center;" colspan="3">

<h2 id="holidayTitle">

</h2>

</td>

</tr>

<tr>

<td style="text-align: left;">

<span id="spanLikesCount" class="voteStyle">

</span>

<img class="rounded" id="imgButtonLike" src="../Pictures/Buttons/ButtonLike\_BW.png" alt="Like" style="width: auto;

height: auto;

border: 0;">

<span id="spanDislikesCount" class="voteStyle">

</span>

<img class="rounded" id="imgButtonDislike" src="../Pictures/Buttons/ButtonDislike\_BW.png" alt="Dislike" style="width: auto;

height: auto;

border: 0;">

</td>

<td style="text-align: center;">

<span id="spanPrice" class="priceStyle">

</span>

</td>

<td style="text-align: left;">

<img class="rounded" id="imgButtonBuy" src="../Pictures/Buttons/ButtonBuy.png" alt="Buy" style="width: auto;

height: auto;

border: 0;" onclick="HolidayInfo\_ButtonBuy\_OnClick();" onmouseenter="HolidayInfo\_ButtonBuy\_OnMouseEnter(this);"

onmouseleave="HolidayInfo\_ButtonBuy\_OnMouseLeave(this);">

</td>

</tr>

<tr>

<td colspan="3">

<span id="spanErrorVote" class="errorStyle">

Error! You must

<a href="Register.html">Register</a>, or

<a href="Login.html">Login</a>, before you can vote, buy or comment a holiday!

</span>

<td>

</tr>

<tr>

<td style="width: 100%;" colspan="3">

<pre class="mb-0" id="preDescription" style="width: 100%;

font-size: 1.15em;">

</pre>

<br>

<pre class="mb-0" id="preTimeLimit">

</pre>

</td>

</tr>

<tr>

<td style="width: 100%;

text-align: center;" colspan="3">

<span id="spanErrorComments" class="errorStyle">

Error! You must

<a href="Register.html">Register</a>, or

<a href="Login.html">Login</a>, before you can vote, buy or comment a holiday!

</span>

</td>

</tr>

<tr>

<td style="width: 100%;

text-align: center;" colspan="3">

<span id="commentsCount">

</span>

</td>

</tr>

<tr>

<td colspan="2" style="padding-left: 10%;

width: 80%;">

<textarea class="form-control" id="taComment" placeholder="Write your comment here..."></textarea>

</td>

<td style="text-align: center;

width: 20%;">

<img class="rounded" id="imgButtonComment" src="../Pictures/Buttons/ButtonComment.png" alt="Comment" style="width: auto;

height: auto;

border: 0;

margin-right: 10%;

vertical-align: top;" onclick="HolidayInfo\_ButtonComment\_OnClick();" onmouseenter="HolidayInfo\_ButtonComment\_OnMouseEnter(this);"

onmouseleave="HolidayInfo\_ButtonComment\_OnMouseLeave(this);">

</td>

</tr>

</table>

<table id="tableComments">

</table>

</div>

</div>

<div class="rightBanner">

<div class="rightBannerMiddleLeft">

<a href="Register.html" id="buttonRegister">

<img class="rounded" src="../Pictures/Buttons/ButtonRegister.png" alt="Register" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonRegister\_OnMouseEnter(this);" onmouseleave="ButtonRegister\_OnMouseLeave(this);">

</a>

<span id="spanUserNames" style="display: none">

User Names

</span>

</div>

<div class="rightBannerMiddleRight">

<a href="Login.html" id="buttonLogin">

<img class="rounded" src="../Pictures/Buttons/ButtonLogIn.png" alt="LogIn" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogin\_OnMouseEnter(this);" onmouseleave="ButtonLogin\_OnMouseLeave(this);">

</a>

<a href="HolidayInfo.html" id="buttonLogout" onclick="Logout()" style="display: none">

<img class="rounded" src="../Pictures/Buttons/ButtonLogOut.png" alt="LogOut" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogout\_OnMouseEnter(this);" onmouseleave="ButtonLogout\_OnMouseLeave(this);">

</a>

</div>

</div>

</body>

</html>

Login.html :

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../Style.css">

<script type="text/javascript" src="../Libraries/Client.js">

</script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>

Holidays - Login

</title>

</head>

<body>

<div class="page">

<div class="mainTopPanel">

<div class="mainTopPanelContainer">

<span class="mainTopPanelButton">

<a href="../Index.html">

<img src="../Pictures/Buttons/ButtonHome.png" alt="Home" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonHome\_OnMouseEnter(this);" onmouseleave="ButtonHome\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="Search.html">

<img src="../Pictures/Buttons/ButtonSearch.png" alt="Search" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonSearch\_OnMouseEnter(this);" onmouseleave="ButtonSearch\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="ContactUs.html">

<img src="../Pictures/Buttons/ButtonContactUs.png" alt="Contact Us" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonContactUs\_OnMouseEnter(this);" onmouseleave="ButtonContactUs\_OnMouseLeave(this);">

</a>

</span>

</span>

</span>

</div>

</div>

<div class="pageContent">

<h1>

Login:

</h1>

<span id="message" style="visibility: hidden">

message

<br>

</span>

<table style="width: 70%;

border: 0px;">

<tr>

<td style="width: 33%;">

Username:

<br>

<input class="form-control" id="inputUsername" type="text" size="33%" maxlength="20" pattern="[A-Za-z0-9.\_-]{4,20}" title="4-20 characters: letters, numbers .\_-"

required>

<br>

<br>

</td>

<td style="width: 1%;">

</td>

<td style="width: 33%;">

Password:

<br>

<input class="form-control" id="inputPassword" type="password" size="33%" maxlength="30" pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{6,30}" title="6-30 characters: At least one capital letter, at least one small letter and at least one number."

required>

<br>

<br>

</td>

</tr>

<tr>

<td colspan="3">

<span style="visibility: hidden">

Adding space

</span>

</td>

</tr>

<tr>

<td colspan="3">

<input class="btn btn-primary" name="buttonLogin" type="submit" value="Login" style="min-width: 100px;

width: 15%;

margin-left: 35%;" onclick="Login()">

</td>

</tr>

</table>

</div>

</div>

<div class="rightBanner">

<div class="rightBannerMiddleLeft">

<a href="Register.html" id="buttonRegister">

<img src="../Pictures/Buttons/ButtonRegister.png" alt="Register" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonRegister\_OnMouseEnter(this);" onmouseleave="ButtonRegister\_OnMouseLeave(this);">

</a>

<span id="spanUserNames" style="display: none">

User Names

</span>

</div>

<div class="rightBannerMiddleRight">

<a href="Login.html" id="buttonLogin">

<img src="../Pictures/Buttons/ButtonLogIn.png" alt="LogIn" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogin\_OnMouseEnter(this);" onmouseleave="ButtonLogin\_OnMouseLeave(this);">

</a>

<a href="Login.html" id="buttonLogout" onclick="Logout()" style="display: none">

<img src="../Pictures/Buttons/ButtonLogOut.png" alt="LogOut" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogout\_OnMouseEnter(this);" onmouseleave="ButtonLogout\_OnMouseLeave(this);">

</a>

</div>

</div>

</body>

</html>

Register.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="../Style.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script type="text/javascript" src="../Libraries/Client.js">

</script>

<title>

Holidays - Register

</title>

</head>

<body>

<div class="page">

<div class="mainTopPanel">

<div class="mainTopPanelContainer">

<span class="mainTopPanelButton">

<a href="../Index.html">

<img src="../Pictures/Buttons/ButtonHome.png" alt="Home" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonHome\_OnMouseEnter(this);" onmouseleave="ButtonHome\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="Search.html">

<img src="../Pictures/Buttons/ButtonSearch.png" alt="Search" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonSearch\_OnMouseEnter(this);" onmouseleave="ButtonSearch\_OnMouseLeave(this);">

</a>

<span class="mainTopPanelButton">

<a href="ContactUs.html">

<img src="../Pictures/Buttons/ButtonContactUs.png" alt="Contact Us" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonContactUs\_OnMouseEnter(this);" onmouseleave="ButtonContactUs\_OnMouseLeave(this);">

</a>

</span>

</span>

</span>

</div>

</div>

<div class="pageContent">

<h1>

Register:

</h1>

<table style="width: 100%;

border: 0px">

<tr>

<td style="width: 33%;">

<span style="color: red">\*</span>Username:

<br>

<input class="form-control" id="inputUsername" type="text" size="33%" maxlength="20" pattern="[A-Za-z0-9.\_-]{4,20}" title="4-20 characters: letters, numbers .\_-"

required>

<br>

<br>

</td>

<td style="width: 33%;">

<span style="color: red">\*</span>First name:

<br>

<input class="form-control" id="inputFirstName" type="text" size="33%" maxlength="44" pattern="[A-Za-z]{4,44}" title="4-44 letters." required>

<br>

<br>

</td>

<td style="width: 33%;">

<span style="color: red">\*</span>Country:

<br>

<input class="form-control" id="inputCountry" type="text" size="33%" maxlength="44" pattern="[A-Za-z]{4,44}" title="4-44 letters" required>

<br>

<br>

</td>

</tr>

<tr>

<td style="width: 33%;">

<span style="color: red">\*</span>Password:

<br>

<input class="form-control" id="inputPassword" type="password" size="33%" maxlength="30" pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{6,30}" title="6-30 characters: At least one capital letter, at least one small letter and at least one number."

required>

<br>

<br>

</td>

<td style="width: 33%;">

Surname:

<br>

<input class="form-control" id="inputSurname" type="text" size="33%" maxlength="44" pattern="[A-Za-z]{4,44}" title="4-44 letters.">

<br>

<br>

</td>

<td style="width: 33%;">

<span style="color: red">\*</span>Town:

<br>

<input class="form-control" id="inputTown" type="text" size="33%" maxlength="44" pattern="[A-Za-z]{4,44}" title="4-44 letters" required>

<br>

<br>

</td>

</tr>

<tr>

<td style="width: 33%;">

<span style="color: red">\*</span>E-mail:

<br>

<input class="form-control" id="inputEmail" type="email" size="33%" maxlength="49" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$" title="(small letters, numbers .\_%+-)@(small letters, numbers .-).(2 or 3 small letters)."

required>

<br>

<br>

</td>

<td style="width: 33%;">

<span style="color: red">\*</span>Last name:

<br>

<input class="form-control" id="inputLastName" type="text" size="33%" maxlength="44" pattern="[A-Za-z]{4,44}" title="4-44 letters." required>

<br>

<br>

</td>

<td style="width: 33%;">

<span style="color: red">\*</span>Address:

<br>

<input class="form-control" id="inputAddress" type="text" size="33%" maxlength="199" pattern="[A-Za-z]{4,199}" title="4-199 letters" required>

<br>

<br>

</td>

</tr>

<tr>

<td style="width: 33%;">

</td>

<td style="width: 33%;">

<span style="visibility: hidden">

Adding space

</span>

</td>

<td style="width: 33%;">

</td>

</tr>

<tr>

<td style="width: 33%;">

</td>

<td style="width: 33%;">

<input class="btn btn-primary" name="buttonRegister" type="button" value="Register" style="min-width: 100px;

margin-left: 20%;

width: 30%;" onclick="Register()">

</td>

<td style="width: 33%;">

</td>

</tr>

</table>

</div>

</div>

<div class="rightBanner">

<div class="rightBannerMiddleLeft">

<a href="Register.html" id="buttonRegister">

<img src="../Pictures/Buttons/ButtonRegister.png" alt="Register" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonRegister\_OnMouseEnter(this);" onmouseleave="ButtonRegister\_OnMouseLeave(this);">

</a>

<span id="spanUserNames" style="display: none">

User Names

</span>

</div>

<div class="rightBannerMiddleRight">

<a href="Login.html" id="buttonLogin">

<img src="../Pictures/Buttons/ButtonLogIn.png" alt="LogIn" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogin\_OnMouseEnter(this);" onmouseleave="ButtonLogin\_OnMouseLeave(this);">

</a>

<a href="Register.html" id="buttonLogout" onclick="Logout()" style="display: none">

<img src="../Pictures/Buttons/ButtonLogOut.png" alt="LogOut" style="width: auto;

height: auto;

border: 0;" onmouseenter="ButtonLogout\_OnMouseEnter(this);" onmouseleave="ButtonLogout\_OnMouseLeave(this);">

</a>

</div>

</div>

</body>

</html>

**JavaScript:**

Client.js :

var Http;

var Url = "http://localhost:8001";

var ArrayHolidays = []; //create an empty array

var Holiday;

var ArrayComments = [];

var CurrentPage = 1;

var MaxHolidaysPerPage = 6;

var IndexPage;

var CurrentImage;

function Register() {

var regExpResult = 1;

var params;

//--------------

var nickname;

var password;

var email;

var firstName;

var surname;

var lastName;

var country;

var town;

var address;

regExpResult = regExpResult & /[A-Za-z0-9.\_-]{4,20}/.test(document.getElementById("inputUsername").value);

regExpResult = regExpResult & /(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{6,30}/.test(document.getElementById("inputPassword").value);

regExpResult = regExpResult & /[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$/.test(document.getElementById("inputEmail").value);

regExpResult = regExpResult & /[A-Za-z]{4,44}/.test(document.getElementById("inputFirstName").value);

if (document.getElementById("inputSurname").value != "") {

regExpResult = regExpResult & /[A-Za-z]{4,44}/.test(document.getElementById("inputSurname").value);

}

regExpResult = regExpResult & /[A-Za-z]{4,44}/.test(document.getElementById("inputLastName").value);

regExpResult = regExpResult & /[A-Za-z]{4,44}/.test(document.getElementById("inputCountry").value);

regExpResult = regExpResult & /[A-Za-z]{4,44}/.test(document.getElementById("inputTown").value);

regExpResult = regExpResult & /[A-Za-z]{4,199}/.test(document.getElementById("inputAddress").value);

if (regExpResult) {

nickname = document.getElementById("inputUsername").value;

password = document.getElementById("inputPassword").value;

email = document.getElementById("inputEmail").value;

firstName = document.getElementById("inputFirstName").value;

surname = document.getElementById("inputSurname").value;

lastName = document.getElementById("inputLastName").value;

country = document.getElementById("inputCountry").value;

town = document.getElementById("inputTown").value;

address = document.getElementById("inputAddress").value;

document.getElementById("inputUsername").value = "";

document.getElementById("inputPassword").value = "";

document.getElementById("inputEmail").value = "";

document.getElementById("inputFirstName").value = "";

document.getElementById("inputSurname").value = "";

document.getElementById("inputLastName").value = "";

document.getElementById("inputCountry").value = "";

document.getElementById("inputTown").value = "";

document.getElementById("inputAddress").value = "";

params = "operation=register&username=" + nickname + "&password=" + password + "&email=" + email + "&firstName=" + firstName + "&surname=" + surname + "&lastName=" + lastName + "&country=" + country + "&town=" + town + "&address=" + address;

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert(Http.responseText);

if (Http.responseText.includes("Success.")) //response contains the phrase "Success."

{

window.location.assign(window.location.hostname + window.location.pathname.substring(0, window.location.pathname.indexOf("/Web/")) + "/Index.html"); //Redirrect to Home page

} else {

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText);

}

}

}

//alert(Http.readyState + " " + Http.status + " " + Http.responseText + ".");

}

Http.send(params);

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function Login() {

var regExpResult = 1;

var params;

//--------------

var nickname;

var password;

regExpResult = regExpResult & /[A-Za-z0-9.\_-]{4,20}/.test(document.getElementById("inputUsername").value);

regExpResult = regExpResult & /(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{6,30}/.test(document.getElementById("inputPassword").value);

if (regExpResult) {

nickname = document.getElementById("inputUsername").value;

password = document.getElementById("inputPassword").value;

document.getElementById("inputUsername").value = "";

document.getElementById("inputPassword").value = "";

params = "operation=login&username=" + nickname + "&password=" + password;

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert(Http.responseText);

if (Http.responseText.includes("Success.")) {

var startIndex;

var endIndex;

var firstName;

var lastName;

startIndex = Http.responseText.indexOf("firstName=");

endIndex = startIndex + Http.responseText.substring(startIndex).indexOf("&");

startIndex += "firstName=".length;

firstName = Http.responseText.substring(startIndex, endIndex);

//alert("firstName:"+firstName);

startIndex = Http.responseText.indexOf("lastName=");

endIndex = Http.responseText.length + 1;

startIndex += "lastName=".length;

lastName = Http.responseText.substring(startIndex, endIndex);

//alert("lastName:"+lastName);

//Write result in cookie

//alert("nickname: " + nickname);

SetCookie("nickname", nickname, 30);

SetCookie("firstName", firstName, 30);

SetCookie("lastName", lastName, 30);

window.location.assign(window.location.hostname + window.location.pathname.substring(0, window.location.pathname.indexOf("/Web/")) + "/Index.html"); //Redirrect to Home page

} else {

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText);

} else {

if (Http.responseText.includes("Fail.")) //response contains the phrase "Fail."

{

document.getElementById("message").innerHTML = "Wrong username or password!<br>";

document.getElementById("message").style = "color: red; font: italic bold 12px/30px Georgia, serif;";

}

}

}

}

//alert(Http.readyState + " " + Http.status + " " + Http.responseText + ".");

}

Http.send(params);

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function Logout() {

SetCookie("nickname", "", -1);

SetCookie("firstName", "", -1);

SetCookie("lastName", "", -1);

}

//--------------------------------------------------------------------------------------------------------------------------------------

function Search() {

var regExpResult = 1;

var searchedValue;

regExpResult = regExpResult & /[A-Za-z0-9А-Яа-я.\_-]{1,70}/.test(document.getElementById("inputSearch").value);

if (regExpResult) {

searchedValue = document.getElementById("inputSearch").value;

params = "operation=search" + "&searchedValue=" + searchedValue;

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert(Http.responseText);

if (Http.responseText.includes("Success.")) {

var startIndex;

var endIndex;

var ID, HolidayName, Destinations, Days, Price, StartDate, EndDate, PicturesURL, Description, ShortDescription, VoteUpCount, VoteDownCount;

var tempResponse;

var counter;

var multiplePictures;

tempResponse = Http.responseText;

ArrayHolidays = []; //Empty the array

while (tempResponse.includes("ID=")) {

startIndex = tempResponse.indexOf("ID=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "ID=".length;

ID = tempResponse.substring(startIndex, endIndex);

//alert("ID:"+ID);

startIndex = tempResponse.indexOf("HolidayName=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "HolidayName=".length;

HolidayName = tempResponse.substring(startIndex, endIndex);

//alert("HolidayName:"+HolidayName);

startIndex = tempResponse.indexOf("Destinations=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Destinations=".length;

Destinations = tempResponse.substring(startIndex, endIndex);

//alert("Destinations:"+Destinations);

startIndex = tempResponse.indexOf("Days=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Days=".length;

Days = tempResponse.substring(startIndex, endIndex);

//alert("Days:"+Days);

startIndex = tempResponse.indexOf("Price=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Price=".length;

Price = tempResponse.substring(startIndex, endIndex);

//alert("Price:"+Price);

startIndex = tempResponse.indexOf("StartDate=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "StartDate=".length;

StartDate = tempResponse.substring(startIndex, endIndex);

//alert("StartDate:"+StartDate);

startIndex = tempResponse.indexOf("EndDate=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "EndDate=".length;

EndDate = tempResponse.substring(startIndex, endIndex);

//alert("EndDate:"+EndDate);

startIndex = tempResponse.indexOf("PicturesURL=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "PicturesURL=".length;

PicturesURL = tempResponse.substring(startIndex, endIndex);

//alert("PicturesURL:"+PicturesURL);

startIndex = tempResponse.indexOf("Description=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Description=".length;

Description = tempResponse.substring(startIndex, endIndex);

//alert("Description:"+Description);

startIndex = tempResponse.indexOf("ShortDescription=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "ShortDescription=".length;

ShortDescription = tempResponse.substring(startIndex, endIndex);

//alert("ShortDescription:"+ShortDescription);

startIndex = tempResponse.indexOf("VoteUpCount=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "VoteUpCount=".length;

VoteUpCount = tempResponse.substring(startIndex, endIndex);

//alert("VoteUpCount:"+VoteUpCount);

startIndex = tempResponse.indexOf("VoteDownCount=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf(";;");

startIndex += "VoteDownCount=".length;

VoteDownCount = tempResponse.substring(startIndex, endIndex);

//alert("VoteDownCount:"+VoteDownCount);

//alert("tempResponse before: " + tempResponse);

tempResponse = tempResponse.substring(endIndex);

//alert("tempResponse after: " + tempResponse);

ArrayHolidays.push({}); //push a new empty object in the array

ArrayHolidays[ArrayHolidays.length - 1].ID = ID;

ArrayHolidays[ArrayHolidays.length - 1].HolidayName = HolidayName;

ArrayHolidays[ArrayHolidays.length - 1].Destinations = Destinations;

ArrayHolidays[ArrayHolidays.length - 1].Days = Days;

ArrayHolidays[ArrayHolidays.length - 1].Price = Price;

ArrayHolidays[ArrayHolidays.length - 1].StartDate = StartDate;

ArrayHolidays[ArrayHolidays.length - 1].EndDate = EndDate;

ArrayHolidays[ArrayHolidays.length - 1].PicturesURL = [];

counter = 0;

multiplePictures = false;

while (PicturesURL.includes(";")) {

multiplePictures = true;

//alert("PicturesURL: " + PicturesURL);

startIndex = 0;

endIndex = PicturesURL.indexOf(";");

//alert("endIndex: " + endIndex);

if (endIndex < startIndex) {

endIndex = PicturesURL.length;

}

ArrayHolidays[ArrayHolidays.length - 1].PicturesURL[counter] = PicturesURL.substring(startIndex, endIndex);

//alert("Picture url: " + ArrayHolidays[ArrayHolidays.length - 1].PicturesURL[counter]);

PicturesURL = PicturesURL.substring(endIndex + 1);

counter++;

}

if (multiplePictures == false) {

startIndex = 0;

endIndex = PicturesURL.length;

ArrayHolidays[ArrayHolidays.length - 1].PicturesURL[counter] = PicturesURL.substring(startIndex, endIndex);

//alert("1");

}

ArrayHolidays[ArrayHolidays.length - 1].Description = Description;

ArrayHolidays[ArrayHolidays.length - 1].ShortDescription = ShortDescription;

ArrayHolidays[ArrayHolidays.length - 1].VoteUpCount = VoteUpCount;

ArrayHolidays[ArrayHolidays.length - 1].VoteDownCount = VoteDownCount;

//alert(ArrayHolidays[ArrayHolidays.length - 1].HolidayName);

}

IndexPage = false;

CurrentPage = 1;

ShowCurrentPage();

} else {

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText);

}

}

}

//alert(Http.readyState + " " + Http.status + " " + Http.responseText + ".");

}

Http.send(params);

} else {

HideHolidays();

}

}

//Events-----------------------------------------------------------------------------------------------------------------------------------------------------

function Index\_OnLoad() {

var firstName;

var lastName;

var params;

//alert("OnLoad()");

if (CheckCookie("firstName")) //cookie exists / user is logged in

{

firstName = GetCookie("firstName");

lastName = GetCookie("lastName");

//alert(firstName + "\n" + lastName);

document.getElementById("buttonRegister").style.display = "none";

document.getElementById("buttonLogin").style.display = "none";

document.getElementById("spanUserNames").style.display = "block";

document.getElementById("spanUserNames").style.width = "80%";

document.getElementById("spanUserNames").innerHTML = "<b>Welcome back,<br>" + firstName + " " + lastName + "</b>";

document.getElementById("buttonLogout").style.display = "block";

} else {

document.getElementById("buttonRegister").style.display = "block";

document.getElementById("buttonLogin").style.display = "block";

document.getElementById("spanUserNames").style.display = "none";

document.getElementById("buttonLogout").style.display = "none";

}

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

params = "operation=holidays";

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert(Http.responseText);

if (Http.responseText.includes("Success.")) {

var startIndex;

var endIndex;

var ID, HolidayName, Destinations, Days, Price, StartDate, EndDate, PicturesURL, Description, ShortDescription, VoteUpCount, VoteDownCount;

var tempResponse;

var counter;

var multiplePictures;

tempResponse = Http.responseText;

//alert("tempResponse: " + tempResponse);

ArrayHolidays = []; //Empty the array

while (tempResponse.includes("ID=")) {

startIndex = tempResponse.indexOf("ID=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "ID=".length;

ID = tempResponse.substring(startIndex, endIndex);

//alert("ID:"+ID);

startIndex = tempResponse.indexOf("HolidayName=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "HolidayName=".length;

HolidayName = tempResponse.substring(startIndex, endIndex);

//alert("HolidayName:"+HolidayName);

startIndex = tempResponse.indexOf("Destinations=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Destinations=".length;

Destinations = tempResponse.substring(startIndex, endIndex);

//alert("Destinations:"+Destinations);

startIndex = tempResponse.indexOf("Days=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Days=".length;

Days = tempResponse.substring(startIndex, endIndex);

//alert("Days:"+Days);

startIndex = tempResponse.indexOf("Price=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Price=".length;

Price = tempResponse.substring(startIndex, endIndex);

//alert("Price:"+Price);

startIndex = tempResponse.indexOf("StartDate=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "StartDate=".length;

StartDate = tempResponse.substring(startIndex, endIndex);

//alert("StartDate:"+StartDate);

startIndex = tempResponse.indexOf("EndDate=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "EndDate=".length;

EndDate = tempResponse.substring(startIndex, endIndex);

//alert("EndDate:"+EndDate);

startIndex = tempResponse.indexOf("PicturesURL=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "PicturesURL=".length;

PicturesURL = tempResponse.substring(startIndex, endIndex);

//alert("PicturesURL:"+PicturesURL);

startIndex = tempResponse.indexOf("Description=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Description=".length;

Description = tempResponse.substring(startIndex, endIndex);

//alert("Description:"+Description);

startIndex = tempResponse.indexOf("ShortDescription=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "ShortDescription=".length;

ShortDescription = tempResponse.substring(startIndex, endIndex);

//alert("ShortDescription:"+ShortDescription);

startIndex = tempResponse.indexOf("VoteUpCount=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "VoteUpCount=".length;

VoteUpCount = tempResponse.substring(startIndex, endIndex);

//alert("VoteUpCount:"+VoteUpCount);

startIndex = tempResponse.indexOf("VoteDownCount=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf(";;");

startIndex += "VoteDownCount=".length;

VoteDownCount = tempResponse.substring(startIndex, endIndex);

//alert("VoteDownCount:"+VoteDownCount);

//alert("tempResponse before: " + tempResponse);

tempResponse = tempResponse.substring(endIndex);

//alert("tempResponse after: " + tempResponse);

ArrayHolidays.push({}); //push a new empty object in the array

ArrayHolidays[ArrayHolidays.length - 1].ID = ID;

ArrayHolidays[ArrayHolidays.length - 1].HolidayName = HolidayName;

ArrayHolidays[ArrayHolidays.length - 1].Destinations = Destinations;

ArrayHolidays[ArrayHolidays.length - 1].Days = Days;

ArrayHolidays[ArrayHolidays.length - 1].Price = Price;

ArrayHolidays[ArrayHolidays.length - 1].StartDate = StartDate;

ArrayHolidays[ArrayHolidays.length - 1].EndDate = EndDate;

ArrayHolidays[ArrayHolidays.length - 1].PicturesURL = [];

counter = 0;

multiplePictures = false;

while (PicturesURL.includes(";")) {

multiplePictures = true;

//alert("PicturesURL: " + PicturesURL);

startIndex = 0;

endIndex = PicturesURL.indexOf(";");

//alert("endIndex: " + endIndex);

if (endIndex < startIndex) {

endIndex = PicturesURL.length;

}

ArrayHolidays[ArrayHolidays.length - 1].PicturesURL[counter] = PicturesURL.substring(startIndex, endIndex);

//alert("Picture url: " + ArrayHolidays[ArrayHolidays.length - 1].PicturesURL[counter]);

PicturesURL = PicturesURL.substring(endIndex + 1);

counter++;

}

if (multiplePictures == false) {

startIndex = 0;

endIndex = PicturesURL.length;

ArrayHolidays[ArrayHolidays.length - 1].PicturesURL[counter] = PicturesURL.substring(startIndex, endIndex);

//alert("1");

}

ArrayHolidays[ArrayHolidays.length - 1].Description = Description;

ArrayHolidays[ArrayHolidays.length - 1].ShortDescription = ShortDescription;

ArrayHolidays[ArrayHolidays.length - 1].VoteUpCount = VoteUpCount;

ArrayHolidays[ArrayHolidays.length - 1].VoteDownCount = VoteDownCount;

//alert(ArrayHolidays[ArrayHolidays.length - 1].HolidayName);

}

IndexPage = true;

CurrentPage = 1;

ShowCurrentPage();

} else {

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText);

}

}

}

//alert(Http.readyState + " " + Http.status + " " + Http.responseText + ".");

}

Http.send(params);

}

//--------------------------------------------------------------------------------------------------------------------------------------

function Search\_OnLoad() {

var firstName;

var lastName;

var params;

//alert("OnLoad()");

if (CheckCookie("firstName")) //cookie exists / user is logged in

{

firstName = GetCookie("firstName");

lastName = GetCookie("lastName");

//alert(firstName + "\n" + lastName);

document.getElementById("buttonRegister").style.display = "none";

document.getElementById("buttonLogin").style.display = "none";

document.getElementById("spanUserNames").style.display = "block";

document.getElementById("spanUserNames").style.width = "80%";

document.getElementById("spanUserNames").innerHTML = "<b>Welcome back,<br>" + firstName + " " + lastName + "</b>";

document.getElementById("buttonLogout").style.display = "block";

} else {

document.getElementById("buttonRegister").style.display = "block";

document.getElementById("buttonLogin").style.display = "block";

document.getElementById("spanUserNames").style.display = "none";

document.getElementById("buttonLogout").style.display = "none";

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function ContactUs\_OnLoad() {

var firstName;

var lastName;

var params;

//alert("OnLoad()");

if (CheckCookie("firstName")) //cookie exists / user is logged in

{

firstName = GetCookie("firstName");

lastName = GetCookie("lastName");

//alert(firstName + "\n" + lastName);

document.getElementById("buttonRegister").style.display = "none";

document.getElementById("buttonLogin").style.display = "none";

document.getElementById("spanUserNames").style.display = "block";

document.getElementById("spanUserNames").style.width = "80%";

document.getElementById("spanUserNames").innerHTML = "<b>Welcome back,<br>" + firstName + " " + lastName + "</b>";

document.getElementById("buttonLogout").style.display = "block";

} else {

document.getElementById("buttonRegister").style.display = "block";

document.getElementById("buttonLogin").style.display = "block";

document.getElementById("spanUserNames").style.display = "none";

document.getElementById("buttonLogout").style.display = "none";

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_OnLoad() {

var firstName;

var lastName;

var nickname;

var params;

var holidayID = "";

var mainTable;

var tempTable;

var tempRow;

var tempCell;

//alert("OnLoad()");

if (CheckCookie("firstName")) //cookie exists / user is logged in

{

firstName = GetCookie("firstName");

lastName = GetCookie("lastName");

nickname = GetCookie("nickname");

//alert(firstName + "\n" + lastName);

document.getElementById("buttonRegister").style.display = "none";

document.getElementById("buttonLogin").style.display = "none";

document.getElementById("spanUserNames").style.display = "block";

document.getElementById("spanUserNames").style.width = "80%";

document.getElementById("spanUserNames").innerHTML = "<b>Welcome back,<br>" + firstName + " " + lastName + "</b>";

document.getElementById("buttonLogout").style.display = "block";

document.getElementById("imgButtonBuy").style.visibility = "visible";

document.getElementById("imgButtonComment").style.visibility = "visible";

document.getElementById("taComment").onclick = function () {};

document.getElementById("spanErrorComments").style.visibility = "hidden";

} else {

document.getElementById("buttonRegister").style.display = "block";

document.getElementById("buttonLogin").style.display = "block";

document.getElementById("spanUserNames").style.display = "none";

document.getElementById("buttonLogout").style.display = "none";

document.getElementById("imgButtonBuy").style.visibility = "hidden";

document.getElementById("imgButtonComment").style.visibility = "hidden";

document.getElementById("taComment").onclick = function () {

document.getElementById("spanErrorComments").style.visibility = "visible";

};

}

if (CheckCookie("HolidayID")) {

holidayID = GetCookie("HolidayID");

//alert("holidayID: " + holidayID);

}

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

params = "operation=holidayInfo" + "&HolidayID=" + holidayID + "&Nickname=" + nickname;

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert("Http.responseText: " + Http.responseText);

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText.substring(Http.responseText.indexOf("Error.")));

} else {

if (Http.responseText.includes("Success.")) {

var startIndex;

var endIndex;

var ID, HolidayName, Destinations, Days, Price, StartDate, EndDate, PicturesURL, Description, ShortDescription, VoteUpCount, VoteDownCount;

var translatedStartDate, translatedEndDate;

var CommentID, Comment, Nickname, HolidayID, DateTime;

var Voted, VoteValue = "";

var tempResponse;

var counter;

var multiplePictures;

var height;

var translatedDateTime;

tempResponse = Http.responseText;

Holiday = {};

startIndex = tempResponse.indexOf("ID=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "ID=".length;

ID = tempResponse.substring(startIndex, endIndex);

//alert("ID:"+ID);

startIndex = tempResponse.indexOf("HolidayName=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "HolidayName=".length;

HolidayName = tempResponse.substring(startIndex, endIndex);

//alert("HolidayName:"+HolidayName);

startIndex = tempResponse.indexOf("Destinations=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Destinations=".length;

Destinations = tempResponse.substring(startIndex, endIndex);

//alert("Destinations:"+Destinations);

startIndex = tempResponse.indexOf("Days=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Days=".length;

Days = tempResponse.substring(startIndex, endIndex);

//alert("Days:"+Days);

startIndex = tempResponse.indexOf("Price=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Price=".length;

Price = tempResponse.substring(startIndex, endIndex);

//alert("Price:"+Price);

startIndex = tempResponse.indexOf("StartDate=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "StartDate=".length;

StartDate = tempResponse.substring(startIndex, endIndex);

//alert("StartDate:"+StartDate);

startIndex = tempResponse.indexOf("EndDate=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "EndDate=".length;

EndDate = tempResponse.substring(startIndex, endIndex);

//alert("EndDate:"+EndDate);

startIndex = tempResponse.indexOf("PicturesURL=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "PicturesURL=".length;

PicturesURL = tempResponse.substring(startIndex, endIndex);

//alert("PicturesURL:"+PicturesURL);

startIndex = tempResponse.indexOf("Description=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Description=".length;

Description = tempResponse.substring(startIndex, endIndex);

//alert("Description:"+Description);

startIndex = tempResponse.indexOf("ShortDescription=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "ShortDescription=".length;

ShortDescription = tempResponse.substring(startIndex, endIndex);

//alert("ShortDescription:"+ShortDescription);

startIndex = tempResponse.indexOf("VoteUpCount=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "VoteUpCount=".length;

VoteUpCount = tempResponse.substring(startIndex, endIndex);

//alert("VoteUpCount:"+VoteUpCount);

startIndex = tempResponse.indexOf("VoteDownCount=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf(";;");

startIndex += "VoteDownCount=".length;

VoteDownCount = tempResponse.substring(startIndex, endIndex);

//alert("VoteDownCount:"+VoteDownCount);

//alert("tempResponse before (holiday): " + tempResponse + "\n\nendIndex: " + endIndex + "tempResponse.length: " + tempResponse.length);

tempResponse = tempResponse.substring(endIndex);

//alert("tempResponse after (holiday): " + tempResponse);

//Holiday processing----------------------------

Holiday.ID = ID;

Holiday.HolidayName = HolidayName;

Holiday.Destinations = Destinations;

Holiday.Days = Days;

Holiday.Price = Price;

Holiday.StartDate = StartDate;

Holiday.EndDate = EndDate;

Holiday.PicturesURL = [];

counter = 0;

multiplePictures = false;

while (PicturesURL.includes(";")) {

multiplePictures = true;

//alert("PicturesURL: " + PicturesURL);

startIndex = 0;

endIndex = PicturesURL.indexOf(";");

//alert("endIndex: " + endIndex);

if (endIndex < startIndex) {

endIndex = PicturesURL.length;

}

Holiday.PicturesURL[counter] = "../" + PicturesURL.substring(startIndex, endIndex);

//alert("Picture url: " + Holiday.PicturesURL[counter]);

PicturesURL = PicturesURL.substring(endIndex + 1);

counter++;

}

if (multiplePictures == false) {

startIndex = 0;

endIndex = PicturesURL.length;

Holiday.PicturesURL[counter] = "../" + PicturesURL.substring(startIndex, endIndex);

//alert("multiplePictures == false");

}

Holiday.Description = Description;

Holiday.ShortDescription = ShortDescription;

Holiday.VoteUpCount = VoteUpCount;

Holiday.VoteDownCount = VoteDownCount;

//alert(Holiday.HolidayName);

//End - Holiday processing----------------------

//alert("tempResponse before (comments): " + tempResponse);

tempResponse = tempResponse.substring(tempResponse.indexOf(";;;Success.") + ";;;Success.".length);

//alert("tempResponse after (comments): " + tempResponse);

ArrayComments = [];

while (tempResponse.includes("ID=")) //there are more comments

{

startIndex = tempResponse.indexOf("ID=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "ID=".length;

CommentID = tempResponse.substring(startIndex, endIndex);

//alert("CommentID: "+CommentID);

startIndex = tempResponse.indexOf("Comment=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Comment=".length;

Comment = tempResponse.substring(startIndex, endIndex);

//alert("Comment: "+Comment);

startIndex = tempResponse.indexOf("Nickname=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "Nickname=".length;

Nickname = tempResponse.substring(startIndex, endIndex);

//alert("Nickname: "+Nickname);

startIndex = tempResponse.indexOf("HolidayID=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

startIndex += "HolidayID=".length;

HolidayID = tempResponse.substring(startIndex, endIndex);

//alert("HolidayID: "+HolidayID);

startIndex = tempResponse.indexOf("DateTime=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf(";;");

startIndex += "DateTime=".length;

DateTime = tempResponse.substring(startIndex, endIndex);

//alert("DateTime: "+DateTime);

/\*Holiday.StartDate - "Tue Jun 06 2017 14:00:00 GMT+0300 (FLE Daylight Time)"

[0-2] - Day of week

[4-6] - Month

[8-9] - Date

[11-14] - Year

[16-23] - Time

[25-32] - Time zone

[34-length] - Daylight time

\*/

/\*Translate datetime---------------\*/

translatedDateTime = "";

translatedDateTime += DateTime.substring(8, 10) + " "; //[8-9] - Date

switch (DateTime.substring(4, 7)) //[4-6] - Month

{

case "Jan":

{

translatedDateTime += "Януари ";

break;

}

case "Feb":

{

translatedDateTime += "Февруари ";

break;

}

case "Mar":

{

translatedDateTime += "Март ";

break;

}

case "Apr":

{

translatedDateTime += "Април ";

break;

}

case "May":

{

translatedDateTime += "Май ";

break;

}

case "Jun":

{

translatedDateTime += "Юни ";

break;

}

case "Jul":

{

translatedDateTime += "Юли ";

break;

}

case "Aug":

{

translatedDateTime += "Август ";

break;

}

case "Sep":

{

translatedDateTime += "Септември ";

break;

}

case "Oct":

{

translatedDateTime += "Октомври ";

break;

}

case "Nov":

{

translatedDateTime += "Ноември ";

break;

}

case "Dec":

{

translatedDateTime += "Декември ";

break;

}

default:

{

translatedDateTime += Holiday.StartDate.substring(4, 7) + " "; //[4-6] - Month

break;

}

}

translatedDateTime += DateTime.substring(11, 15) + ", "; //[11-14] - Year

translatedDateTime += DateTime.substring(16, 24); //[16-23] - Time

//alert("tempResponse before (comments): " + tempResponse);

tempResponse = tempResponse.substring(endIndex + 2); //+2 for the ;; delimiter

//alert("tempResponse after (comments): " + tempResponse);

ArrayComments.push({}); //push an empty object in the array

ArrayComments[ArrayComments.length - 1].ID = CommentID;

ArrayComments[ArrayComments.length - 1].Comment = Comment;

ArrayComments[ArrayComments.length - 1].Nickname = Nickname;

ArrayComments[ArrayComments.length - 1].HolidayID = HolidayID;

ArrayComments[ArrayComments.length - 1].DateTime = DateTime;

ArrayComments[ArrayComments.length - 1].translatedDateTime = translatedDateTime;

//alert("ArrayComments[ArrayComments.length - 1].Comment: " + ArrayComments[ArrayComments.length - 1].Comment);

}

//alert("tempResponse before (comments): " + tempResponse);

tempResponse = tempResponse.substring(tempResponse.indexOf(";;;") + ";;;".length);

//alert("tempResponse after (comments): " + tempResponse);

startIndex = tempResponse.indexOf("Voted=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + tempResponse.substring(startIndex).indexOf(".");

}

startIndex += "Voted=".length;

Voted = tempResponse.substring(startIndex, endIndex);

//alert("Voted: "+Voted);

if (Voted == "true") {

startIndex = tempResponse.indexOf("VoteValue=");

endIndex = startIndex + tempResponse.substring(startIndex).indexOf(";;");

if (endIndex < startIndex) {

endIndex = tempResponse.length;

}

startIndex += "VoteValue=".length;

VoteValue = tempResponse.substring(startIndex, endIndex);

//alert("VoteValue: "+VoteValue);

}

Holiday.Voted = Voted;

Holiday.VoteValue = VoteValue;

//Load content on page-----------------------------

document.getElementById("holidayTitle").innerHTML = Holiday.ShortDescription;

document.getElementById("spanPrice").innerHTML = "<b>" + Holiday.Price + " лв</b>";

CurrentImage = 0;

HolidayInfo\_ShowCurrentImage();

document.getElementById("preDescription").innerHTML = Holiday.Description;

/\*Holiday.StartDate - "Tue Jun 06 2017 14:00:00 GMT+0300 (FLE Daylight Time)"

[0-2] - Day of week

[4-6] - Month

[8-9] - Date

[11-14] - Year

[16-23] - Time

[25-32] - Time zone

[34-length] - Daylight time

\*/

/\*Translate datetime---------------\*/

translatedStartDate = "";

translatedEndDate = "";

translatedStartDate += Holiday.StartDate.substring(8, 10) + " "; //[8-9] - Date

switch (Holiday.StartDate.substring(4, 7)) //[4-6] - Month

{

case "Jan":

{

translatedStartDate += "Януари ";

break;

}

case "Feb":

{

translatedStartDate += "Февруари ";

break;

}

case "Mar":

{

translatedStartDate += "Март ";

break;

}

case "Apr":

{

translatedStartDate += "Април ";

break;

}

case "May":

{

translatedStartDate += "Май ";

break;

}

case "Jun":

{

translatedStartDate += "Юни ";

break;

}

case "Jul":

{

translatedStartDate += "Юли ";

break;

}

case "Aug":

{

translatedStartDate += "Август ";

break;

}

case "Sep":

{

translatedStartDate += "Септември ";

break;

}

case "Oct":

{

translatedStartDate += "Октомври ";

break;

}

case "Nov":

{

translatedStartDate += "Ноември ";

break;

}

case "Dec":

{

translatedStartDate += "Декември ";

break;

}

default:

{

translatedStartDate += Holiday.StartDate.substring(4, 7) + " "; //[4-6] - Month

break;

}

}

translatedStartDate += Holiday.StartDate.substring(11, 15) + ", "; //[11-14] - Year

translatedStartDate += Holiday.StartDate.substring(16, 24); //[16-23] - Time

translatedEndDate += Holiday.EndDate.substring(8, 10) + " "; //[8-9] - Date

switch (Holiday.EndDate.substring(4, 7)) //[4-6] - Month

{

case "Jan":

{

translatedEndDate += "Януари ";

break;

}

case "Feb":

{

translatedEndDate += "Февруари ";

break;

}

case "Mar":

{

translatedEndDate += "Март ";

break;

}

case "Apr":

{

translatedEndDate += "Април ";

break;

}

case "May":

{

translatedEndDate += "Май ";

break;

}

case "Jun":

{

translatedEndDate += "Юни ";

break;

}

case "Jul":

{

translatedEndDate += "Юли ";

break;

}

case "Aug":

{

translatedEndDate += "Август ";

break;

}

case "Sep":

{

translatedEndDate += "Септември ";

break;

}

case "Oct":

{

translatedEndDate += "Октомври ";

break;

}

case "Nov":

{

translatedEndDate += "Ноември ";

break;

}

case "Dec":

{

translatedEndDate += "Декември ";

break;

}

default:

{

translatedEndDate += Holiday.EndDate.substring(4, 7) + " "; //[4-6] - Month

break;

}

}

translatedEndDate += Holiday.EndDate.substring(11, 15) + ", "; //[11-14] - Year

translatedEndDate += Holiday.EndDate.substring(16, 24); //[16-23] - Time

/\*End - Translate datetime---------\*/

document.getElementById("preTimeLimit").innerHTML = "Офертата е валидна от " + translatedStartDate + " до " + translatedEndDate + ".<br>";

if (window.innerWidth > window.innerHeight) {

document.getElementById("imgHoliday").style.height = (window.innerHeight \* 0.7) + "px";

document.getElementById("imgHoliday").style.width = "auto";

} else {

if (window.innerWidth < window.innerHeight) {

document.getElementById("imgHoliday").style.width = (window.innerWidth \* 0.7) + "px";

document.getElementById("imgHoliday").style.height = "auto";

}

}

document.getElementById("spanLikesCount").innerHTML = "&nbsp;" + Holiday.VoteUpCount + "&nbsp;";

document.getElementById("spanDislikesCount").innerHTML = "&nbsp;" + Holiday.VoteDownCount + "&nbsp;";

HolidayInfo\_UpdateVote();

//alert("ArrayComments.length: " + ArrayComments.length);

document.getElementById("commentsCount").innerHTML = ArrayComments.length + " comments:";

document.getElementById("taComment").style.width = (window.innerWidth \* 0.45) + "px";;

document.getElementById("taComment").style.height = (window.innerHeight \* 0.17) + "px";

//Load comments----------------------------

mainTable = document.getElementById("tableComments");

mainTable.style.borderRadius = "15px";

for (counter = 0; counter < ArrayComments.length; counter++) {

tempTable = document.createElement("TABLE");

tempTable.style.border = "3px solid darkgrey";

tempTable.style.width = "100%";

tempTable.style.backgroundColor = "lightgrey";

tempTable.style.borderRadius = "15px";

tempRow = tempTable.insertRow(0);

tempCell = tempRow.insertCell(0);

tempCell.innerHTML = ArrayComments[counter].Nickname;

tempCell.style.backgroundColor = "lightgrey";

tempCell.style.borderBottom = "3px solid darkgrey";

tempCell.style.borderRight = "3px solid darkgrey";

tempCell.style.width = "20%";

tempCell.style.textAlign = "center";

tempCell.style.borderTopLeftRadius = "15px";

tempCell.style.borderTopRightRadius = "15px";

tempCell = tempRow.insertCell(1);

tempCell.rowSpan = "2";

tempCell.innerHTML = ArrayComments[counter].Comment;

tempCell.style.backgroundColor = "white";

tempCell.style.borderTopRightRadius = "15px";

tempCell.style.borderBottomRightRadius = "15px";

tempRow = tempTable.insertRow(1);

tempCell = tempRow.insertCell(0);

tempCell.innerHTML = ArrayComments[counter].translatedDateTime;

tempCell.style.backgroundColor = "lightgrey";

tempCell.style.borderRight = "3px solid darkgrey";

tempCell.style.width = "20%";

tempCell.style.borderBottomLeftRadius = "15px";

tempCell.style.borderBottomRightRadius = "15px";

mainTable.insertRow(0).insertCell(0).appendChild(tempTable);

}

//End - Load comments----------------------

//End - Load content on page-----------------------

}

}

}

//alert(Http.readyState + " " + Http.status + " " + Http.responseText + ".");

}

Http.send(params);

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_ButtonBuy\_OnClick() {

var params;

var nickname;

nickname = GetCookie("nickname");

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

params = "operation=buyHoliday" + "&Description=" + Holiday.ShortDescription + "&TotalPrice=" + Holiday.Price + "&Nickname=" + nickname + "&HolidayID=" + Holiday.ID;

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert("Http.responseText: " + Http.responseText);

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText.substring(Http.responseText.indexOf("Error.")));

} else {

if (Http.responseText.includes("Success.")) {

alert("Holiday purchased successfully!");

}

}

}

}

Http.send(params);

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_ButtonComment\_OnClick() {

var params;

var nickname;

var comment;

nickname = GetCookie("nickname");

comment = document.getElementById("taComment").value;

document.getElementById("taComment").value = "";

//alert("comment: " + comment);

if (comment != "") {

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

params = "operation=commentHoliday" + "&Comment=" + comment + "&Nickname=" + nickname + "&HolidayID=" + Holiday.ID;

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert("Http.responseText: " + Http.responseText);

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText.substring(Http.responseText.indexOf("Error.")));

} else {

if (Http.responseText.includes("Success.")) {

window.location.assign(window.location.hostname + window.location.pathname); //Refresh page

}

}

}

}

Http.send(params);

} else {

alert("You must write a comment first!");

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_OnResize() {

if (window.innerWidth > window.innerHeight) {

document.getElementById("imgHoliday").style.height = (window.innerHeight \* 0.7) + "px";

document.getElementById("imgHoliday").style.width = "auto";

} else {

if (window.innerWidth < window.innerHeight) {

document.getElementById("imgHoliday").style.width = (window.innerWidth \* 0.7) + "px";

document.getElementById("imgHoliday").style.height = "auto";

}

}

document.getElementById("taComment").style.width = (window.innerWidth \* 0.45) + "px";

document.getElementById("taComment").style.height = (window.innerHeight \* 0.17) + "px";

}

//--------------------------------------------------------------------------------------------------------------------------------------

function ButtonPreviousPage\_OnClick() {

CurrentPage--;

ShowCurrentPage();

}

//--------------------------------------------------------------------------------------------------------------------------------------

function ButtonNextPage\_OnClick() {

CurrentPage++;

ShowCurrentPage();

}

//End - Events-----------------------------------------------------------------------------------------------------------------------------------------------

//Cookies----------------------------------------------------------------------------------------------------------------------------------------------------

function SetCookie(cookieName, cookieValue, expireDays) {

var date;

var expires;

var path = "/";

date = new Date();

date.setTime(date.getTime() + (expireDays \* 24 \* 60 \* 60 \* 1000)); //expire time is in miliseconds (\*24\*60\*60\*1000 converts it to days)

expires = "expires=" + date.toUTCString(); //toGMTString() is possible too

document.cookie = cookieName + "=" + cookieValue + ";" + expires + ";path=" + path;

//alert("Cookies:" + document.cookie);

}

//--------------------------------------------------------------------------------------------------------------------------------------

function GetCookie(cookieName) {

var name;

var decodedCookie;

var cookieArray;

name = cookieName + "=";

decodedCookie = decodeURIComponent(document.cookie);

cookieArray = decodedCookie.split(';');

for (var i = 0; i < cookieArray.length; i++) {

var cookieValue;

cookieValue = cookieArray[i];

while (cookieValue.charAt(0) == ' ') {

cookieValue = cookieValue.substring(1);

}

if (cookieValue.indexOf(name) == 0) {

return cookieValue.substring(name.length, cookieValue.length);

}

}

return "";

}

//--------------------------------------------------------------------------------------------------------------------------------------

function CheckCookie(cookieValue) {

var temp;

temp = GetCookie(cookieValue);

//alert("Temp: " + temp);

if (temp != "") {

return true;

} else {

return false;

}

}

//End - Cookies----------------------------------------------------------------------------------------------------------------------------------------------

//Additional functions---------------------------------------------------------------------------------------------------------------------------------------

function ShowCurrentPage() {

var allPages;

var repeatCount;

if (ArrayHolidays.length != 0) {

if (ArrayHolidays.length % MaxHolidaysPerPage == 0) {

allPages = ArrayHolidays.length / MaxHolidaysPerPage;

} else {

allPages = Math.trunc(ArrayHolidays.length / MaxHolidaysPerPage) + 1;

}

if (CurrentPage == allPages) //last page

{

repeatCount = ArrayHolidays.length % MaxHolidaysPerPage;

} else {

repeatCount = MaxHolidaysPerPage;

}

//alert("allPages: " + allPages + "\n" + "repeatCount: " + repeatCount);

for (var i = 0; i < repeatCount; i++) //show the holidays on the page

{

if (IndexPage) {

document.getElementById("imgHoliday" + (i + 1)).src = ArrayHolidays[((CurrentPage - 1) \* MaxHolidaysPerPage) + i].PicturesURL[0];

} else {

document.getElementById("imgHoliday" + (i + 1)).src = "../" + ArrayHolidays[((CurrentPage - 1) \* MaxHolidaysPerPage) + i].PicturesURL[0];

}

document.getElementById("imgHoliday" + (i + 1)).style.visibility = "visible";

document.getElementById("priceHoliday" + (i + 1)).innerHTML = ArrayHolidays[((CurrentPage - 1) \* MaxHolidaysPerPage) + i].Price + " лв";

document.getElementById("priceHoliday" + (i + 1)).style.visibility = "visible";

document.getElementById("shortDescriptionHoliday" + (i + 1)).innerHTML = ArrayHolidays[((CurrentPage - 1) \* MaxHolidaysPerPage) + i].ShortDescription;

document.getElementById("shortDescriptionHoliday" + (i + 1)).style.visibility = "visible";

/\* The same as the code below with a little difference

document.getElementById("tableHoliday" + (i + 1)).onclick = (function()

{

var currentI = i + 1;

return function()

{

ProcessHoliday(currentI);

}

})();\*/

//Neater implementation

document.getElementById("tableHoliday" + (i + 1)).onclick = function (currentI) {

return function () {

ProcessHoliday(currentI);

}

}(i + 1);

/\*The (i + 1) invokes the function(currentI) with the parameter being the current i,

triggering it to return the function in which a copied

value is in the closure, no longer connected to i\*/

}

for (var i = repeatCount; i < MaxHolidaysPerPage; i++) //hide the unnecessary holidays on the page

{

document.getElementById("imgHoliday" + (i + 1)).style.visibility = "hidden";

document.getElementById("priceHoliday" + (i + 1)).style.visibility = "hidden";

document.getElementById("shortDescriptionHoliday" + (i + 1)).style.visibility = "hidden";

document.getElementById("tableHoliday" + (i + 1)).onclick = function () {};

}

document.getElementById("imgPreviousPage").style.visibility = "visible";

if (CurrentPage > 1) //there is previous page

{

if (IndexPage) {

document.getElementById("imgPreviousPage").src = "Pictures/Buttons/ButtonArrowLeft.png";

} else {

document.getElementById("imgPreviousPage").src = "../Pictures/Buttons/ButtonArrowLeft.png";

}

document.getElementById("imgPreviousPage").onclick = ButtonPreviousPage\_OnClick;

document.getElementById("imgPreviousPage").onmouseenter = function () {

ButtonPreviousPage\_OnMouseEnter(this)

};

document.getElementById("imgPreviousPage").onmouseleave = function () {

ButtonPreviousPage\_OnMouseLeave(this)

};

} else {

if (IndexPage) {

document.getElementById("imgPreviousPage").src = "Pictures/Buttons/ButtonArrowLeft-Grey.png";

} else {

document.getElementById("imgPreviousPage").src = "../Pictures/Buttons/ButtonArrowLeft-Grey.png";

}

document.getElementById("imgPreviousPage").onclick = function () {};

document.getElementById("imgPreviousPage").onmouseenter = function () {};

document.getElementById("imgPreviousPage").onmouseleave = function () {};

}

document.getElementById("imgNextPage").style.visibility = "visible";

if (CurrentPage < allPages) //there is next page

{

if (IndexPage) {

document.getElementById("imgNextPage").src = "Pictures/Buttons/ButtonArrowRight.png";

} else {

document.getElementById("imgNextPage").src = "../Pictures/Buttons/ButtonArrowRight.png";

}

document.getElementById("imgNextPage").onclick = ButtonNextPage\_OnClick;

document.getElementById("imgNextPage").onmouseenter = function () {

ButtonNextPage\_OnMouseEnter(this)

};

document.getElementById("imgNextPage").onmouseleave = function () {

ButtonNextPage\_OnMouseLeave(this)

};

} else {

if (IndexPage) {

document.getElementById("imgNextPage").src = "Pictures/Buttons/ButtonArrowRight-Grey.png";

} else {

document.getElementById("imgNextPage").src = "../Pictures/Buttons/ButtonArrowRight-Grey.png";

}

document.getElementById("imgNextPage").onclick = function () {};

document.getElementById("imgNextPage").onmouseenter = function () {};

document.getElementById("imgNextPage").onmouseleave = function () {};

}

document.getElementById("spanCurrentPage").style.visibility = "visible";

document.getElementById("spanCurrentPage").innerHTML = "<b>Page: " + CurrentPage + "/" + allPages + "</b>";

} else {

HideHolidays();

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HideHolidays() {

for (var i = 0; i < MaxHolidaysPerPage; i++) //hide all holidays on the page

{

document.getElementById("imgHoliday" + (i + 1)).style.visibility = "hidden";

document.getElementById("priceHoliday" + (i + 1)).style.visibility = "hidden";

document.getElementById("shortDescriptionHoliday" + (i + 1)).style.visibility = "hidden";

}

document.getElementById("imgPreviousPage").style.visibility = "hidden";

document.getElementById("imgNextPage").style.visibility = "hidden";

document.getElementById("spanCurrentPage").style.visibility = "hidden";

}

//--------------------------------------------------------------------------------------------------------------------------------------

function ProcessHoliday(indexOnPage) {

//alert("indexOnPage: " + indexOnPage);

var arrayIndex;

var id;

arrayIndex = ((CurrentPage - 1) \* MaxHolidaysPerPage) + indexOnPage - 1; //-1 because 1st array index is 0, but 1st index of holiday on page is 1

//alert("arrayIndex: " + arrayIndex);

id = ArrayHolidays[arrayIndex].ID;

//alert("id: " + id);

SetCookie("HolidayID", id, 7);

//alert("window.location.pathname: " + window.location.pathname);

if (window.location.pathname.includes("Index.html")) {

window.location.assign(window.location.hostname + window.location.pathname.substring(0, window.location.pathname.indexOf("Index.html")) + "Web/HolidayInfo.html"); //Redirrect to HolidayInfo page

} else {

window.location.assign(window.location.hostname + window.location.pathname.substring(0, (window.location.pathname.indexOf("Web/") + "Web/".length)) + "HolidayInfo.html"); //Redirrect to HolidayInfo page

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_ShowCurrentImage() {

document.getElementById("imgHoliday").src = Holiday.PicturesURL[CurrentImage];

if (CurrentImage == Holiday.PicturesURL.length - 1) //there are no more images after

{

document.getElementById("imgNextImage").src = "../Pictures/Buttons/ButtonArrowRight-Grey.png";

document.getElementById("imgNextImage").onclick = function () {};

document.getElementById("imgNextImage").onmouseenter = function () {};

document.getElementById("imgNextImage").onmouseleave = function () {};

} else {

document.getElementById("imgNextImage").src = "../Pictures/Buttons/ButtonArrowRight.png";

document.getElementById("imgNextImage").onclick = HolidayInfo\_ShowNextImage;

document.getElementById("imgNextImage").onmouseenter = function () {

ButtonNextPage\_OnMouseEnter(this)

};

document.getElementById("imgNextImage").onmouseleave = function () {

ButtonNextPage\_OnMouseLeave(this)

};

}

if (CurrentImage == 0) //there are no more images before

{

document.getElementById("imgPreviousImage").src = "../Pictures/Buttons/ButtonArrowLeft-Grey.png";

document.getElementById("imgPreviousImage").onclick = function () {};

document.getElementById("imgPreviousImage").onmouseenter = function () {};

document.getElementById("imgPreviousImage").onmouseleave = function () {};

} else {

document.getElementById("imgPreviousImage").src = "../Pictures/Buttons/ButtonArrowLeft.png";

document.getElementById("imgPreviousImage").onclick = HolidayInfo\_ShowPreviousImage;

document.getElementById("imgPreviousImage").onmouseenter = function () {

ButtonPreviousPage\_OnMouseEnter(this)

};

document.getElementById("imgPreviousImage").onmouseleave = function () {

ButtonPreviousPage\_OnMouseLeave(this)

};

}

document.getElementById("spanCurrentImage").innerHTML = "Image " + (CurrentImage + 1) + " / " + Holiday.PicturesURL.length + " ";

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_ShowNextImage() {

CurrentImage++;

HolidayInfo\_ShowCurrentImage();

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_ShowPreviousImage() {

CurrentImage--;

HolidayInfo\_ShowCurrentImage();

}

//--------------------------------------------------------------------------------------------------------------------------------------

function HolidayInfo\_UpdateVote() {

//alert("Holiday.Voted: " + Holiday.Voted);

if (CheckCookie("nickname")) //user is logged in

{

if (Holiday.Voted == "true") {

//alert("Holiday.VotedValue: " + Holiday.VoteValue);

if (Holiday.VoteValue == 1) {

document.getElementById("imgButtonLike").src = "../Pictures/Buttons/ButtonLike.png";

document.getElementById("imgButtonLike").onclick = function () {};

document.getElementById("imgButtonDislike").src = "../Pictures/Buttons/ButtonDislike\_BW.png";

document.getElementById("imgButtonDislike").onclick = function () {

Vote(-1);

};

} else {

if (Holiday.VoteValue == -1) {

document.getElementById("imgButtonLike").src = "../Pictures/Buttons/ButtonLike\_BW.png";

document.getElementById("imgButtonLike").onclick = function () {

Vote(1);

};

document.getElementById("imgButtonDislike").src = "../Pictures/Buttons/ButtonDislike.png";

document.getElementById("imgButtonDislike").onclick = function () {};

}

}

document.getElementById("imgButtonLike").onmouseenter = function () {};

document.getElementById("imgButtonLike").onmouseleave = function () {};

document.getElementById("imgButtonDislike").onmouseenter = function () {};

document.getElementById("imgButtonDislike").onmouseleave = function () {};

} else {

if (Holiday.Voted == "false") {

document.getElementById("imgButtonLike").src = "../Pictures/Buttons/ButtonLike\_BW.png";

document.getElementById("imgButtonLike").onclick = function () {

Vote(1);

};

document.getElementById("imgButtonDislike").src = "../Pictures/Buttons/ButtonDislike\_BW.png";

document.getElementById("imgButtonDislike").onclick = function () {

Vote(-1);

};

document.getElementById("imgButtonLike").onmouseenter = function () {

HolidayInfo\_ButtonLike\_OnMouseEnter(this)

};

document.getElementById("imgButtonLike").onmouseleave = function () {

HolidayInfo\_ButtonLike\_OnMouseLeave(this)

};

document.getElementById("imgButtonDislike").onmouseenter = function () {

HolidayInfo\_ButtonDislike\_OnMouseEnter(this)

};

document.getElementById("imgButtonDislike").onmouseleave = function () {

HolidayInfo\_ButtonDislike\_OnMouseLeave(this)

};

}

}

} else //user is nor logged in

{

document.getElementById("imgButtonLike").onclick = function () {

document.getElementById("spanErrorVote").style.visibility = "visible";

};

document.getElementById("imgButtonDislike").onclick = function () {

document.getElementById("spanErrorVote").style.visibility = "visible";

};

}

}

//--------------------------------------------------------------------------------------------------------------------------------------

function Vote(voteValue) {

var params;

var holidayID;

var nickname;

holidayID = GetCookie("HolidayID");

nickname = GetCookie("nickname");

//alert("voteValue: " + voteValue);

Http = new XMLHttpRequest();

Http.open("POST", Url, true);

if (Holiday.Voted == "false") {

params = "operation=vote" + "&VoteValue=" + voteValue + "&HolidayID=" + holidayID + "&Nickname=" + nickname;

} else {

if (Holiday.Voted == "true") {

params = "operation=changeVote" + "&VoteValue=" + voteValue + "&HolidayID=" + holidayID + "&Nickname=" + nickname;

}

}

//Send the proper header information along with the request

Http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

Http.onreadystatechange = function () { //Call a function when the state changes.

if (Http.readyState == 4 && Http.status == 200) {

//alert("Http.responseText: " + Http.responseText);

if (Http.responseText.includes("Error.")) //response contains the phrase "Error."

{

alert(Http.responseText.substring(Http.responseText.indexOf("Error.")));

} else {

if (Http.responseText.includes("Success.")) //response contains the phrase "Success."

{

if (Holiday.Voted == "false") {

Holiday.Voted = "true";

}

Holiday.VoteValue = voteValue;

HolidayInfo\_UpdateVote();

}

}

}

}

Http.send(params);

}

//End - Additional functions---------------------------------------------------------------------------------------------------------------------------------

//Animations-------------------------------------------------------------------------------------------------------------------------------------------------

function ButtonHome\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonHome\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonHome\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonHome\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonHome.png";

} else {

button.src = "../Pictures/Buttons/ButtonHome.png";

}

}

//-------------------------------------------------------------------------------------------------------

function ButtonSearch\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonSearch\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonSearch\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonSearch\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonSearch.png";

} else {

button.src = "../Pictures/Buttons/ButtonSearch.png";

}

}

//-------------------------------------------------------------------------------------------------------

function ButtonContactUs\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonContactUs\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonContactUs\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonContactUs\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonContactUs.png";

} else {

button.src = "../Pictures/Buttons/ButtonContactUs.png";

}

}

//-------------------------------------------------------------------------------------------------------

function ButtonRegister\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonRegister\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonRegister\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonRegister\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonRegister.png";

} else {

button.src = "../Pictures/Buttons/ButtonRegister.png";

}

}

//-------------------------------------------------------------------------------------------------------

function ButtonLogin\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonLogIn\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonLogIn\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonLogin\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonLogIn.png";

} else {

button.src = "../Pictures/Buttons/ButtonLogIn.png";

}

}

//-------------------------------------------------------------------------------------------------------

function ButtonLogout\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonLogOut\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonLogOut\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonLogout\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonLogOut.png";

} else {

button.src = "../Pictures/Buttons/ButtonLogOut.png";

}

}

//-------------------------------------------------------------------------------------------------------

function ButtonPreviousPage\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonArrowLeft\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonArrowLeft\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonPreviousPage\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonArrowLeft.png";

} else {

button.src = "../Pictures/Buttons/ButtonArrowLeft.png";

}

}

//-------------------------------------------------------------------------------------------------------

function ButtonNextPage\_OnMouseEnter(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonArrowRight\_H.png";

} else {

button.src = "../Pictures/Buttons/ButtonArrowRight\_H.png";

}

}

//-------------------------------------------------------------------

function ButtonNextPage\_OnMouseLeave(button) {

if (window.location.pathname.includes("Index.html")) {

button.src = "Pictures/Buttons/ButtonArrowRight.png";

} else {

button.src = "../Pictures/Buttons/ButtonArrowRight.png";

}

}

//-------------------------------------------------------------------------------------------------------

function HolidayInfo\_ButtonBuy\_OnMouseEnter(button) {

button.src = "../Pictures/Buttons/ButtonBuy\_H.png";

}

//-------------------------------------------------------------------

function HolidayInfo\_ButtonBuy\_OnMouseLeave(button) {

button.src = "../Pictures/Buttons/ButtonBuy.png";

}

//-------------------------------------------------------------------------------------------------------

function HolidayInfo\_ButtonComment\_OnMouseEnter(button) {

button.src = "../Pictures/Buttons/ButtonComment\_H.png";

}

//-------------------------------------------------------------------

function HolidayInfo\_ButtonComment\_OnMouseLeave(button) {

button.src = "../Pictures/Buttons/ButtonComment.png";

}

//-------------------------------------------------------------------------------------------------------

function HolidayInfo\_ButtonLike\_OnMouseEnter(button) {

button.src = "../Pictures/Buttons/ButtonLike.png";

}

//-------------------------------------------------------------------

function HolidayInfo\_ButtonLike\_OnMouseLeave(button) {

button.src = "../Pictures/Buttons/ButtonLike\_BW.png";

}

//-------------------------------------------------------------------------------------------------------

function HolidayInfo\_ButtonDislike\_OnMouseEnter(button) {

button.src = "../Pictures/Buttons/ButtonDislike.png";

}

//-------------------------------------------------------------------

function HolidayInfo\_ButtonDislike\_OnMouseLeave(button) {

button.src = "../Pictures/Buttons/ButtonDislike\_BW.png";

}

//-------------------------------------------------------------------------------------------------------

//End - Animations-------------------------------------------------------------------------------------------------------------------------------------------

**5.2.1 Сървърна чст:**

Server.js :

//Libraries ------------------------------------------------------------------------

var HTTP = require('http'),

URL = require('url'),

MySQL = require('mysql');

//-Libraries -----------------------------------------------------------------------

//Global Variables -----------------------------------------------------------------

var Port = 8001;

var Pool = MySQL.createPool({

connectionLimit: 10,

host: 'localhost',

user: 'root',

password: 'BlackSwan93!',

database: 'mydb'

});

//-Global Variables ----------------------------------------------------------------

HTTP.createServer(

function (request, response) {

var operation;

var startIndex;

var endIndex;

var headers = request.headers;

var method = request.method;

var url = request.url;

var body = [];

var params = [];

request.on('error', function (err) {

console.error(err);

response.statusCode = 400;

response.end();

}).on('data', function (chunk) {

//console.log("Chunk: " + chunk + ".");

body.push(chunk);

}).on('end', function () {

body = Buffer.concat(body).toString();

//console.log(body);

console.log("Request received!");

// At this point, we have the headers, method, url and body, and can now

// do whatever we need to in order to respond to this request.

response.on('error', function (err) {

console.error(err);

});

//console.log("Body: " + body + "\n");

startIndex = body.indexOf("operation=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

//console.log("startIndex: " + startIndex);

//console.log("endIndex: " + endIndex);

if (endIndex <= startIndex) {

endIndex = body.length;

}

startIndex += "operation=".length;

operation = body.substring(startIndex, endIndex);

//console.log("Operation: " + operation);

switch (operation) {

case "register":

{

var counter = 0;

//console.log("Operation register!");

startIndex = body.indexOf("username=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "username=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("username: " + params[counter]);

counter++;

startIndex = body.indexOf("password=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "password=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("password: " + params[counter]);

counter++;

startIndex = body.indexOf("email=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "email=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("email: " + params[counter]);

counter++;

startIndex = body.indexOf("firstName=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "firstName=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("firstName: " + params[counter]);

counter++;

startIndex = body.indexOf("surname=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "surname=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("surname: " + params[counter]);

counter++;

startIndex = body.indexOf("lastName=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "lastName=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("lastName: " + params[counter]);

counter++;

startIndex = body.indexOf("country=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "country=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("country: " + params[counter]);

counter++;

startIndex = body.indexOf("town=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "town=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("town: " + params[counter]);

counter++;

startIndex = body.indexOf("address=");

endIndex = body.length + 1;

startIndex += "address=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("address: " + params[counter]);

//counter++;

break;

}

//------------------------------------------------------------------------------------------------

case "login":

{

var counter = 0;

//console.log("Operation register!");

startIndex = body.indexOf("username=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

startIndex += "username=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("username: " + params[counter]);

counter++;

startIndex = body.indexOf("password=");

endIndex = body.length + 1;

startIndex += "password=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("password: " + params[counter]);

break;

}

//------------------------------------------------------------------------------------------------

case "holidays":

{

//console.log("Operation holidays!");

break;

}

//------------------------------------------------------------------------------------------------

case "search":

{

//console.log("Operation search!");

var counter = 0;

startIndex = body.indexOf("searchedValue=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "searchedValue=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("startIndex: " + startIndex + "\nendIndex: " + endIndex + "\nparams[counter]: " + params[counter]);

//console.log("searchedValue: " + params[counter]);

counter++;

break;

}

//------------------------------------------------------------------------------------------------

case "holidayInfo":

{

//console.log("Operation holidayInfo!");

var counter = 0;

//Index 0

startIndex = body.indexOf("HolidayID=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "HolidayID=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("startIndex: " + startIndex + "\nendIndex: " + endIndex + "\nparams[counter]: " + params[counter]);

//console.log("HolidayID: " + params[counter]);

counter++;

//Index 1

startIndex = body.indexOf("Nickname=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "Nickname=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("startIndex: " + startIndex + "\nendIndex: " + endIndex + "\nparams[counter]: " + params[counter]);

//console.log("Nickname: " + params[counter]);

counter++;

break;

}

//------------------------------------------------------------------------------------------------

case "vote":

{

//console.log("Operation holidayInfo!");

var counter = 0;

//Index 0

startIndex = body.indexOf("VoteValue=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "VoteValue=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("VoteValue: " + params[counter]);

counter++;

//Index 1

startIndex = body.indexOf("HolidayID=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "HolidayID=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("HolidayID: " + params[counter]);

counter++;

//Index 2

startIndex = body.indexOf("Nickname=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "Nickname=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("Nickname: " + params[counter]);

counter++;

break;

}

//------------------------------------------------------------------------------------------------

case "changeVote":

{

//console.log("Operation holidayInfo!");

var counter = 0;

//Index 0

startIndex = body.indexOf("VoteValue=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "VoteValue=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("VoteValue: " + params[counter]);

counter++;

//Index 1

startIndex = body.indexOf("HolidayID=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "HolidayID=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("HolidayID: " + params[counter]);

counter++;

//Index 2

startIndex = body.indexOf("Nickname=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "Nickname=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("Nickname: " + params[counter]);

counter++;

break;

}

//------------------------------------------------------------------------------------------------

case "buyHoliday":

{

//console.log("Operation buyHoliday!");

var counter = 0;

//Index 0

startIndex = body.indexOf("Description=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "Description=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("Description: " + params[counter]);

counter++;

//Index 1

startIndex = body.indexOf("TotalPrice=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "TotalPrice=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("TotalPrice: " + params[counter]);

counter++;

//Index 2

startIndex = body.indexOf("Nickname=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "Nickname=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("Nickname: " + params[counter]);

counter++;

//Index 3

startIndex = body.indexOf("HolidayID=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "HolidayID=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("HolidayID: " + params[counter]);

counter++;

break;

}

//------------------------------------------------------------------------------------------------

case "commentHoliday":

{

//console.log("Operation buyHoliday!");

var counter = 0;

//Index 0

startIndex = body.indexOf("Comment=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "Comment=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("Comment: " + params[counter]);

counter++;

//Index 1

startIndex = body.indexOf("Nickname=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "Nickname=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("Nickname: " + params[counter]);

counter++;

//Index 2

startIndex = body.indexOf("HolidayID=");

endIndex = startIndex + body.substring(startIndex).indexOf("&");

if (endIndex < startIndex) {

endIndex = startIndex + body.substring(startIndex).length;

}

startIndex += "HolidayID=".length;

params[counter] = body.substring(startIndex, endIndex);

//console.log("HolidayID: " + params[counter]);

counter++;

break;

}

}

ConnectToDatabase(response, operation, params);

});

}).listen(Port);

console.log("\nServer initialized!\n");

//Additional functions-------------------------------------------------------------------------------------------------------------------------------------

function ConnectToDatabase(response, operation, params) {

Pool.getConnection(

function (err, connection) {

var sql;

if (err) {

console.log('Error connecting to Db!');

return;

} else {

console.log('Connection to database established!\nConnected as id: ' + connection.threadId + '!');

switch (operation) {

case "register":

{

//console.log(params[0] + "\n" + params[1] + "\n" + params[2] + "\n" + params[3] + "\n" + params[4] + "\n" + params[5] + "\n" + params[6] + "\n" + params[7] + "\n" + params[8] + "\n");

if (params[4] == "") {

connection.query("INSERT INTO users (Nickname, Password, Role, Email, FirstName, LastName, Country, Town, Address) VALUES (?, ?, 'Member', ?, ?, ?, ?, ?, ?)", //? is for avoiding SQL injection attack

[params[0], params[1], params[2], params[3], params[5], params[6], params[7], params[8]], //params replacing the ?

function (error, results, fields) {

if (error) {

console.log('Error adding user to database!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

} else {

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Success.");

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

} else {

connection.query("INSERT INTO users (Nickname, Password, Role, Email, FirstName, Surname, LastName, Country, Town, Address) VALUES (?, ?, 'Member', ?, ?, ?, ?, ?, ?, ?)", //? is for avoiding SQL injection attack

[params[0], params[1], params[2], params[3], params[4], params[5], params[6], params[7], params[8]], //params replacing the ?

function (error, results, fields) {

if (error) {

console.log('Error adding user to database!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

} else {

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Success.");

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

}

break;

}

//------------------------------------------------------------------------------------------------

case "login":

{

//console.log(params[0] + "\n" + params[1]);

connection.query("SELECT \* FROM users WHERE Nickname=? AND Password=?", //? is for avoiding SQL injection attack

[params[0], params[1]], //params replacing the ?

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

} else {

var responseMessage;

if (results.length == 0) //Username or password are invalid

{

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

responseMessage = "Fail.";

response.end(responseMessage);

} else //Username and password are correct

{

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

responseMessage = "Success.firstName=" + results[0]["FirstName"] + "&lastName=" + results[0]["LastName"];

response.end(responseMessage);

}

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

break;

}

//------------------------------------------------------------------------------------------------

case "holidays":

{

connection.query("SELECT \* FROM holidays WHERE StartDate < NOW() AND NOW() < EndDate",

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

} else {

var responseMessage = "";

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

//Cycle------------------------------------------------------------------------------------------------------

responseMessage += "Success.";

for (var i = 0; i < results.length; i++) {

responseMessage += "ID=" + results[i]["ID"] +

"&HolidayName=" + results[i]["HolidayName"] +

"&Destinations=" + results[i]["Destinations"] +

"&Days=" + results[i]["Days"] +

"&Price=" + results[i]["Price"] +

"&StartDate=" + results[i]["StartDate"] +

"&EndDate=" + results[i]["EndDate"] +

"&PicturesURL=" + results[i]["PicturesURL"] +

"&Description=" + results[i]["Description"] +

"&ShortDescription=" + results[i]["ShortDescription"] +

"&VoteUpCount=" + results[i]["VoteUpCount"] +

"&VoteDownCount=" + results[i]["VoteDownCount"] + ";;";

//console.log(results[i]["HolidayName"] + "\n");

}

response.end(responseMessage);

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

break;

}

//------------------------------------------------------------------------------------------------

case "search":

{

params[0] = '%' + params[0] + '%'; //% means 0 to any count of symbols

connection.query("SELECT \* FROM holidays WHERE (HolidayName LIKE ? OR Destinations LIKE ? OR Description LIKE ? OR ShortDescription LIKE ?) AND StartDate < NOW() AND NOW() < EndDate", //? is for avoiding SQL injection attack

[params[0], params[0], params[0], params[0]], //params replacing the ?

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

} else {

var responseMessage = "";

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

//Cycle------------------------------------------------------------------------------------------------------

responseMessage += "Success.";

for (var i = 0; i < results.length; i++) {

responseMessage += "ID=" + results[i]["ID"] +

"&HolidayName=" + results[i]["HolidayName"] +

"&Destinations=" + results[i]["Destinations"] +

"&Days=" + results[i]["Days"] +

"&Price=" + results[i]["Price"] +

"&StartDate=" + results[i]["StartDate"] +

"&EndDate=" + results[i]["EndDate"] +

"&PicturesURL=" + results[i]["PicturesURL"] +

"&Description=" + results[i]["Description"] +

"&ShortDescription=" + results[i]["ShortDescription"] +

"&VoteUpCount=" + results[i]["VoteUpCount"] +

"&VoteDownCount=" + results[i]["VoteDownCount"] + ";;";

//console.log(results[i]["HolidayName"] + "\n");

}

response.end(responseMessage);

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

break;

}

//------------------------------------------------------------------------------------------------

case "holidayInfo":

{

connection.query("SELECT \* FROM holidays WHERE ID = ?", //? is for avoiding SQL injection attack

[params[0]], //params replacing the ?

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

} else {

var responseMessage = "";

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

//Cycle------------------------------------------------------------------------------------------------------

responseMessage += "Success.";

//console.log("\nresults.length: " + results.length);

for (var i = 0; i < results.length; i++) {

responseMessage += "ID=" + results[i]["ID"] +

"&HolidayName=" + results[i]["HolidayName"] +

"&Destinations=" + results[i]["Destinations"] +

"&Days=" + results[i]["Days"] +

"&Price=" + results[i]["Price"] +

"&StartDate=" + results[i]["StartDate"] +

"&EndDate=" + results[i]["EndDate"] +

"&PicturesURL=" + results[i]["PicturesURL"] +

"&Description=" + results[i]["Description"] +

"&ShortDescription=" + results[i]["ShortDescription"] +

"&VoteUpCount=" + results[i]["VoteUpCount"] +

"&VoteDownCount=" + results[i]["VoteDownCount"] + ";;";

//console.log("Description: " + results[i]["Description"].substring(0, 300) + "\n");

}

responseMessage += ";;;";

response.write(responseMessage);

connection.query("SELECT \* FROM comments WHERE HolidayID = ?", //? is for avoiding SQL injection attack

[params[0]], //params replacing the ?

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.end("Error. Error code: " + error.code);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

} else {

responseMessage = "";

//Cycle------------------------------------------------------------------------------------------------------

responseMessage += "Success.";

for (var i = 0; i < results.length; i++) {

responseMessage += "ID=" + results[i]["ID"] +

"&Comment=" + results[i]["Comment"] +

"&Nickname=" + results[i]["Nickname"] +

"&HolidayID=" + results[i]["HolidayID"] +

"&DateTime=" + results[i]["DateTime"] + ";;";

//console.log("results[i][\"Comment\"]: " + results[i]["Comment"] + "\n");

}

responseMessage += ";;;";

response.write(responseMessage);

connection.query("SELECT CAST(VoteValue AS UNSIGNED) AS VoteValue FROM votes WHERE HolidayID = ? AND Nickname = ?", //? is for avoiding SQL injection attack

[params[0], params[1]], //params replacing the ?

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.end("Error. Error code: " + error.code);

} else {

responseMessage = "";

//Cycle------------------------------------------------------------------------------------------------------

responseMessage += "Success.";

if (results.length > 0) {

//console.log("\nresults[0]['VoteValue']: " + (results[0]["VoteValue"] - 1));

responseMessage += "Voted=true" + "&VoteValue=" + (results[0]["VoteValue"] - 1);

} else {

responseMessage += "Voted=false.";

}

responseMessage += ";;";

response.end(responseMessage);

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

}

});

}

});

break;

}

//------------------------------------------------------------------------------------------------

case "vote":

{

var SQL;

if (params[0] == 1) //VoteValue

{

SQL = "INSERT INTO votes (VoteValue, HolidayID, Nickname) VALUES (b'10', ?, ?)";

} else {

if (params[0] == -1) {

SQL = "INSERT INTO votes (VoteValue, HolidayID, Nickname) VALUES (b'00', ?, ?)";

}

}

connection.query(SQL, [params[1], params[2]],

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

} else {

var responseMessage = "";

var SQL;

if (params[0] == 1) //VoteValue

{

SQL = "UPDATE holidays SET VoteUpCount = VoteUpCount + 1 WHERE ID = ?";

} else {

if (params[0] == -1) {

SQL = "UPDATE holidays SET VoteDownCount = VoteDownCount + 1 WHERE ID = ?";

}

}

connection.query(SQL,

params[1],

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

} else {

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

responseMessage += "Success.";

response.end(responseMessage);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

}

});

}

});

break;

}

//------------------------------------------------------------------------------------------------

case "changeVote":

{

var SQL;

if (params[0] == 1) //VoteValue

{

SQL = "UPDATE votes SET VoteValue = b'10' WHERE HolidayID = ? AND Nickname = ?";

} else {

if (params[0] == -1) {

SQL = "UPDATE votes SET VoteValue = b'00' WHERE HolidayID = ? AND Nickname = ?";

}

}

connection.query(SQL, [params[1], params[2]],

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

} else {

var SQL;

if (params[0] == 1) //VoteValue

{

SQL = "UPDATE holidays SET VoteUpCount = VoteUpCount + 1 WHERE ID = ?";

} else {

if (params[0] == -1) {

SQL = "UPDATE holidays SET VoteDownCount = VoteDownCount + 1 WHERE ID = ?";

}

}

connection.query(SQL,

params[1],

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

} else {

var responseMessage = "";

var SQL;

if (params[0] == 1) //VoteValue

{

SQL = "UPDATE holidays SET VoteDownCount = VoteDownCount - 1 WHERE ID = ?";

} else {

if (params[0] == -1) {

SQL = "UPDATE holidays SET VoteUpCount = VoteUpCount - 1 WHERE ID = ?";

}

}

connection.query(SQL,

params[1],

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

} else {

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

responseMessage += "Success.";

response.end(responseMessage);

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

}

});

}

});

}

});

break;

}

//------------------------------------------------------------------------------------------------

case "buyHoliday":

{

connection.query("INSERT INTO orders (Description, TotalPrice, DateOrdered, Nickname, HolidayID) VALUES (?, ?, NOW(), ?, ?)", [params[0], params[1], params[2], params[3]],

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

} else {

var responseMessage = "";

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

responseMessage += "Success.";

response.end(responseMessage);

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

break;

}

//------------------------------------------------------------------------------------------------

case "commentHoliday":

{

connection.query("INSERT INTO comments (Comment, Nickname, HolidayID, DateTime) VALUES (?, ?, ?, NOW())", [params[0], params[1], params[2]],

function (error, results, fields) {

if (error) {

console.log('Error in sending query!\nError code: ' + error.code + "\n");

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

response.end("Error. Error code: " + error.code);

} else {

var responseMessage = "";

response.writeHead(200, {

"Content-Type": "text/plain",

"Access-Control-Allow-Origin": "\*"

});

responseMessage += "Success.";

response.end(responseMessage);

}

console.log("Response sent!");

connection.release();

console.log("Connection to database released!\n\n");

});

break;

}

//------------------------------------------------------------------------------------------------

default:

{

connection.release();

console.log("Connection to database released!\n\n");

break;

}

}

}

});

}

//End - Additional functions-------------------------------------------------------------------------------------------------------------------------------

Database:

myDB-dump\_holiday\_planner.sql

-- MySQL dump 10.13 Distrib 5.7.12, for Win32 (AMD64)

--

-- Host: 127.0.0.1 Database: mydb

-- ------------------------------------------------------

-- Server version 5.7.16-log

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

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

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

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `comments`

--

DROP TABLE IF EXISTS `comments`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `comments` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`Comment` varchar(500) NOT NULL,

`Nickname` varchar(30) NOT NULL,

`HolidayID` int(11) NOT NULL,

`DateTime` datetime NOT NULL,

PRIMARY KEY (`ID`),

KEY `Nickname\_idx` (`Nickname`),

KEY `HolidayID\_idx` (`HolidayID`),

CONSTRAINT `NN` FOREIGN KEY (`Nickname`) REFERENCES `users` (`Nickname`) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT `commentsHolidayID` FOREIGN KEY (`HolidayID`) REFERENCES `holidays` (`ID`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `comments`

--

LOCK TABLES `comments` WRITE;

/\*!40000 ALTER TABLE `comments` DISABLE KEYS \*/;

INSERT INTO `comments` VALUES (1,'Test comment','test',3,'2018-01-27 18:08:45'),(2,'Another comment.','test',3,'2018-08-04 23:04:45'),(3,'Още един коментар.','test',3,'2018-08-08 17:47:29'),(4,'Много готино място за почивка!','test',2,'2018-08-08 17:51:53'),(5,'Много хубаво местенце за отдих! Всичко е перфектно! :)','I.Ivanov',1,'2018-08-08 17:53:25'),(6,'Идеално място за почивка далеч от шумния град.','I.Ivanov',3,'2018-08-08 17:55:20');

/\*!40000 ALTER TABLE `comments` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `holidays`

--

DROP TABLE IF EXISTS `holidays`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `holidays` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`HolidayName` varchar(45) NOT NULL,

`Destinations` varchar(200) NOT NULL,

`Days` int(11) NOT NULL,

`Price` double NOT NULL,

`StartDate` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`EndDate` datetime NOT NULL,

`PicturesURL` varchar(3000) NOT NULL DEFAULT 'Pictures\\Holidays\\HolidayNotAvailable.jpg',

`ShortDescription` varchar(150) NOT NULL,

`Description` varchar(10000) NOT NULL,

`VoteUpCount` int(11) NOT NULL DEFAULT '0',

`VoteDownCount` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `holidays`

--

LOCK TABLES `holidays` WRITE;

/\*!40000 ALTER TABLE `holidays` DISABLE KEYS \*/;

INSERT INTO `holidays` VALUES (1,'Chateau Windy Hills','Chateau Windy Hills\*\* край Сливенски минерални бани',2,95,'2018-03-03 14:00:00','2018-08-30 23:59:59','Pictures\\Holidays\\Chateau Windy Hills\\1.jpeg;Pictures\\Holidays\\Chateau Windy Hills\\2.png;Pictures\\Holidays\\Chateau Windy Hills\\3.jpeg;Pictures\\Holidays\\Chateau Windy Hills\\4.png;Pictures\\Holidays\\Chateau Windy Hills\\5.png;Pictures\\Holidays\\Chateau Windy Hills\\6.png;Pictures\\Holidays\\Chateau Windy Hills\\7.jpeg;Pictures\\Holidays\\Chateau Windy Hills\\8.png;Pictures\\Holidays\\Chateau Windy Hills\\9.jpeg;Pictures\\Holidays\\Chateau Windy Hills\\10.png;Pictures\\Holidays\\Chateau Windy Hills\\11.jpeg;Pictures\\Holidays\\Chateau Windy Hills\\12.png','Почивка край Сливенски минерални бани. Две нощувки със закуски и вечери за двама, от Chateau Windy Hills\*\*',' Мечтаете си за романтично бягство от града? Chateau Windy Hills\*\* ви очаква до Сливенски минерални бани, с възможност да опитате отбрани сортове вино от собствената му изба!\nОфертата включва още разходка с екскурзовод из производствените зали, цеха за бутилиране на вината, избата за съхранението им и атракционната кула.\n\n С предварителна резервация на: 0886 467 444, 0886 977 933.\n Един ваучер е за двама, настанени в двойна стая.\n Вечерята е по двустепенно сет меню (основно ястие и десерт), плюс чаша вино (бяло или червено).\n Офертата включва още разходка с екскурзовод из производствените зали, цеха за бутилиране на вината, избата за съхранението им и атракционната кула.\n\n Дете до 3 ненавършени години се настанява безплатно.\n За дете от 3 до 12 ненавършени години, настанено на допълнително легло, се доплащат на рецепция 10лв на ден, на база нощувка със закуска.\n За дете на и над 12г или трети възрастен на допълнително легло се доплащат на рецепция 20лв на ден, на база нощувка със закуска.\n\n Chateau Windy Hills е разположено на 12км западно от Сливен на главен път София - Бургас в непосредствена близост до Сливенски минерални бани. Разполага с масив от 420 декара лозови насаждения от висококачествен френски посадъчен материал от сортовете \"Мерло\", \"Каберне Совиньон\", \"Сира\" и \"Александрийски Мускат\". Избата е проектирана за преработка на 500 тона грозде при затворен технологичен цикъл.\n\n Сградата е разположена в центъра на лозовия масив, а от нейната атракционна кула се открива прекрасна гледка към масива и околностите. Комплекс \"Windy Hills\" разполага с ресторант със зимна и лятна градина с по 80 места. В специално обособената дегустационна зала с 25 места нашите гости могат да дегустират уникалния вкус на вината.\n\n В четиризвездната хотелска част в Chateau Windy Hills\*\* има 11 уютни двойни стаи за двама, където гостите ни биха могли да прекарат истински релаксиращ уикенд. За посетителите се организира разходка с екскурзовод (технолог винар) из производствените зали, цеха за бутилиране на вината, избата за съхранението им, атракционната кула и други.',2,0),(2,'Хотел Снежанка','Хотел Снежанка\*\*\* в центъра на Пампорово',5,215,'2018-03-03 14:00:00','2018-08-30 23:59:59','Pictures\\Holidays\\Снежанка\\1.jpeg;Pictures\\Holidays\\Снежанка\\2.jpeg;Pictures\\Holidays\\Снежанка\\3.jpeg;Pictures\\Holidays\\Снежанка\\4.jpeg;Pictures\\Holidays\\Снежанка\\5.jpeg;Pictures\\Holidays\\Снежанка\\6.jpeg;Pictures\\Holidays\\Снежанка\\7.jpeg;Pictures\\Holidays\\Снежанка\\8.jpeg;Pictures\\Holidays\\Снежанка\\9.jpeg;Pictures\\Holidays\\Снежанка\\10.jpeg;Pictures\\Holidays\\Снежанка\\11.jpeg;Pictures\\Holidays\\Снежанка\\12.jpeg;Pictures\\Holidays\\Снежанка\\13.jpeg;Pictures\\Holidays\\Снежанка\\14.jpeg;Pictures\\Holidays\\Снежанка\\15.jpeg;Pictures\\Holidays\\Снежанка\\16.jpeg','Прохладно лято в Пампорово! 2, 3 или 5 нощувки на база All Inclusive, от Хотел Снежанка\*\*\*',' Отдайте се на прохлада и релакс през горещото лято в Пампорово! Хотел Снежанка\*\*\* ви очаква със страхотно предложение в центъра на любимия ви курорт!\n\n 5 нощувки на база All Inclusive - за 215лв на човек.\n\n All Inclusive пакетът включва:\n\n • Закуска от 7:30 - 10:00ч - блок маса в основния ресторант на хотела с включени топли напитки;\n • Обяд от 12:30 - 14:00ч - блок маса в основния ресторант на хотела с включени напитки - кафе и чай, безалкохолно, минерална вода и българско вино бяло/червено и бира; Обядът може да бъде \"сух пакет\" при заявка на рецепция един ден предварително;\n • Следобедна закуска от 16:00 - 17:00ч - основния ресторант - пици, сандвичи, дребни сладки, кроасанчета и кекс;\n • Вечеря от 19:00 - 21:30ч - блок маса в основния ресторант на хотела - кафе и чай, безалкохолно, минерална вода и български алкохолни напитки, вино, бира;\n • Напитки от 14:00 - 22:00ч - в основния ресторанта на хотела - кафе, чай, безалкохолни напитки, минерална вода, български алкохолни напитки, вино бяло/червено и бира.\n\n Изхранването е на блок маса при повече от 20 гости.\n\n С предварителна резервация на: 03095/83 16, 0885 838 410, 02/878 010 и 0887 08 2050.\n\n Един ваучер е за 1 човек, настанен в двойна стая при двама настанени пълноплащащи.\nОфертата включва още ползване на вътрешен басейн, сауна, Wi-fi, застраховка, туристическа такса, 9% ДДС.\n\n Дете до 6 ненавършени години се на допълнително легло в двойна стая с двама възрастни се настанява безплатно.\n За дете от 6 до 12 ненавършени години на допълнително легло в двойна стая се доплащат на рецепция 16лв/ден, на база пакета.\n За дете от 2 до 12 ненавършени години на редовно легло в двойна стая се доплащат на рецепция 50% от стойността на ваучера за съответния вариант.\n Възрастен (на и над 12г) на допълнително легло в двойна стая доплаща на рецепция 75% от стойността на ваучера за съответния вариант.\n\n За настаняване в апартамент е необходимо да се грабнат 3 ваучера.\n\n\n Хотел Снежанка\*\*\* e елегантен хотел с идеално местоположение в центъра на ски курорта Пампорово. Хотелът разполага с 67 двойни стаи и 86 апартамента с общ капацитет oколо 250 гости. Гостите могат да релаксират с питие в модерния 80 местен лоби-бар.\n\n Ресторантът на хотела предлага уютна атмосфера и разнообразие от ястия с капацитет от 280 места. Може да се насладите на закуска на блок-масата или в комфорта на вашата стая. 80 местната механа е с автентичен родопски интериор. Комплексът включва още дискотека, магазин, детски кът, игрална зала, компютърен клуб, безплатен паркинг и Wi-Fi интернет на своите посетители.',2,0),(3,'Комплекс Старосел','Комплекс Старосел\*\*\* в с. Старосел',1,135,'2018-03-03 14:00:00','2018-08-30 23:59:59','Pictures\\Holidays\\Старосел\\1.jpeg;Pictures\\Holidays\\Старосел\\2.jpeg;Pictures\\Holidays\\Старосел\\3.jpeg;Pictures\\Holidays\\Старосел\\4.jpeg;Pictures\\Holidays\\Старосел\\5.jpeg;Pictures\\Holidays\\Старосел\\6.jpeg;Pictures\\Holidays\\Старосел\\7.jpeg;Pictures\\Holidays\\Старосел\\8.jpeg;Pictures\\Holidays\\Старосел\\9.jpeg;Pictures\\Holidays\\Старосел\\10.jpeg;Pictures\\Holidays\\Старосел\\11.jpeg;Pictures\\Holidays\\Старосел\\12.jpeg;Pictures\\Holidays\\Старосел\\13.jpeg;Pictures\\Holidays\\Старосел\\14.jpeg;Pictures\\Holidays\\Старосел\\15.jpeg;Pictures\\Holidays\\Старосел\\16.jpeg','SPA релакс край Хисаря! Нощувка със закуска за трима, от Комплекс Старосел\*\*\*, с. Старосел',' Чували ли сте за винарска изба Старосел, за останките от тракийска култура по тези места? Дайте си глътка релакс в хотел Старосел\*\*\*, а може и да отскочите до Хисаря да си налеете лековита вода!\n\n Делничен пакет: 1 нощувка със закуска в апартамент за трима възрастни, или двама възрастни и две деца до 12г - за 135лв.\n\n Офертата включва още:\n • открит и закрит минерален басейн;\n • джакузи с минерална вода;\n • парна баня, сауна, релакс зона;\n • паркинг, интернет;\n • винен тур.\n\n Офертата важи за настаняване от Неделя до Четвъртък.\n С предварителна резервация на: 0897 870 908.\n\n Анулации, промени по резервации и неявяванане:\n • Повече от 10 дни от датата на пристигане - 0% от стойността на стаите;\n • Между 10-тия и до ден преди пристигането - 50% от стойността на стаите;\n • Анулиране на деня или непристигане се заплаща 100% от стойността на стаите.\n\n Един ваучер е за трима възрастни или двама възрастни и две деца до 12 ненавършени години, настанени в апартамент в апартхотел Старосел.\n\n Офертата включва още:\n • открит и закрит минерален басейн;\n • джакузи с минерална вода;\n • парна баня, сауна, релакс зона;\n • паркинг, интернет;\n • винен тур.\n\n Деца до 2 ненавършени години се настаняват безплатно.\n За едно дете от 2 до 12 ненавършени години се доплащат на рецепция 25лв на ден, на база пакета.\n За второ дете от 2 до 12 ненавършени години в студио се доплащат на рецепция 15лв на ден, на база пакета.\n\n Апартамент \"Приятели\" е с две спални за настаняване на четирима възрастни или двама възрастни и две деца.\n\n За домашни любимци се заплащат 15лв на нощувка, с предварително одобрение на любимеца от ръководството на хотела. Моля да уведомите предварително, ако възнамерявате да посетите хотела с домашен любимец.\n\n\n Комплексът за винен и SPA туризъм Старосел\*\*\* разполага общо с 42 места за настаняване. Интериорът на хотела е изпълнен с естествените материали и ръчно изработените мебели и тъкани, които представят магията на българския фолклор. Съчетанието на топлите цветове с природните извивки на дървото предлагат на гостите едно незабравимо преживяване.\n\n Винарска изба Старосел е разположена в подножието на древен тракийски храм. Разкритите в и около храма винарски съдове и пособия за правене на вино категорично доказват, че траките са преработвали грозде и пиели вино в околностите на село Старосел още V-VІ век преди Христа.\n\n Наред с безценните исторически паметници това е още едно наследство, с което като българи трябва да се гордеем. Направените от екипа на д-р Китов археологически разкрития с право нареждат Старосел между най-старите винени тероари в света. ',2,0);

/\*!40000 ALTER TABLE `holidays` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `orders`

--

DROP TABLE IF EXISTS `orders`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `orders` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`Description` varchar(200) NOT NULL,

`TotalPrice` double NOT NULL,

`DateOrdered` datetime NOT NULL,

`Nickname` varchar(30) NOT NULL,

`HolidayID` int(11) NOT NULL,

PRIMARY KEY (`ID`),

KEY `UserName\_idx` (`Nickname`),

KEY `ordersHolidayID\_idx` (`HolidayID`),

CONSTRAINT `UserName` FOREIGN KEY (`Nickname`) REFERENCES `users` (`Nickname`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `ordersHolidayID` FOREIGN KEY (`HolidayID`) REFERENCES `holidays` (`ID`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `orders`

--

LOCK TABLES `orders` WRITE;

/\*!40000 ALTER TABLE `orders` DISABLE KEYS \*/;

INSERT INTO `orders` VALUES (1,'SPA релакс край Хисаря! Нощувка със закуска за трима, от Комплекс Старосел\*\*\*, с. Старосел',135,'2018-08-08 17:32:36','I.Ivanov',3),(2,'Прохладно лято в Пампорово! 2, 3 или 5 нощувки на база All Inclusive, от Хотел Снежанка\*\*\*',215,'2018-08-08 17:33:24','I.Ivanov',2),(3,'Почивка край Сливенски минерални бани. Две нощувки със закуски и вечери за двама, от Chateau Windy Hills\*\*',95,'2018-08-08 17:33:58','test',1),(4,'SPA релакс край Хисаря! Нощувка със закуска за трима, от Комплекс Старосел\*\*\*, с. Старосел',135,'2018-08-08 17:47:57','test',3);

/\*!40000 ALTER TABLE `orders` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `roles`

--

DROP TABLE IF EXISTS `roles`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `roles` (

`Role` varchar(30) NOT NULL,

`EditSite` bit(1) NOT NULL DEFAULT b'0',

`WriteComments` bit(1) NOT NULL DEFAULT b'0',

`VoteHolidays` bit(1) NOT NULL DEFAULT b'0',

PRIMARY KEY (`Role`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `roles`

--

LOCK TABLES `roles` WRITE;

/\*!40000 ALTER TABLE `roles` DISABLE KEYS \*/;

INSERT INTO `roles` VALUES ('Admin','','',''),('Member','\0','','');

/\*!40000 ALTER TABLE `roles` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `users`

--

DROP TABLE IF EXISTS `users`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `users` (

`Nickname` varchar(30) NOT NULL,

`Password` varchar(30) NOT NULL,

`Role` varchar(30) NOT NULL,

`Email` varchar(50) NOT NULL,

`FirstName` varchar(45) NOT NULL,

`Surname` varchar(45) DEFAULT NULL,

`LastName` varchar(45) NOT NULL,

`Country` varchar(45) NOT NULL,

`Town` varchar(45) NOT NULL,

`Address` varchar(200) NOT NULL,

PRIMARY KEY (`Nickname`),

KEY `\_idx` (`Role`),

CONSTRAINT `Role` FOREIGN KEY (`Role`) REFERENCES `roles` (`Role`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `users`

--

LOCK TABLES `users` WRITE;

/\*!40000 ALTER TABLE `users` DISABLE KEYS \*/;

INSERT INTO `users` VALUES ('I.Ivanov','Admin123','Admin','ivanov@gmail.com','Iskren','I.','Ivanov','Bulgaria','Gabrovo','ul. \"Studentska\" 71, 211'),('test','testQ1','Member','test@asd.bg','test','test','test','test','test','test');

/\*!40000 ALTER TABLE `users` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `votes`

--

DROP TABLE IF EXISTS `votes`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `votes` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`HolidayID` int(11) NOT NULL,

`VoteValue` bit(2) NOT NULL DEFAULT b'1',

`Nickname` varchar(30) NOT NULL,

PRIMARY KEY (`ID`),

KEY `Nickname\_idx` (`Nickname`),

KEY `votesHolidayID\_idx` (`HolidayID`),

CONSTRAINT `HolidayID` FOREIGN KEY (`HolidayID`) REFERENCES `holidays` (`ID`) ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT `Nickname` FOREIGN KEY (`Nickname`) REFERENCES `users` (`Nickname`) ON DELETE NO ACTION ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=21 DEFAULT CHARSET=utf8;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `votes`

--

LOCK TABLES `votes` WRITE;

/\*!40000 ALTER TABLE `votes` DISABLE KEYS \*/;

INSERT INTO `votes` VALUES (6,5,'\0','test'),(8,6,'\0','test'),(9,7,'','test'),(10,8,'\0','test'),(11,1,'','test'),(12,2,'','test'),(13,3,'','test'),(14,4,'\0','test'),(16,1,'','I.Ivanov'),(17,2,'','I.Ivanov'),(18,3,'','I.Ivanov'),(19,4,'\0','I.Ivanov'),(20,6,'','I.Ivanov');

/\*!40000 ALTER TABLE `votes` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Dumping events for database 'mydb'

--

--

-- Dumping routines for database 'mydb'

--

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

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

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

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

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;