МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ЧЕРНИГОВСКИЙ НАЦИОНАЛЬННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ И КОМПЬЮТЕРНЫХ СИСТЕМ

ОТЧЕТ

о прохождении учебной практики

по информационным технологиям

на тему: Веб-приложение «Информационная система «Биатлон»»

Исполнители:

ст.гр. КИ-132 Хропатый А.Н.

Пода В.С.

Небожак А.Н.

Руководитель:

ассистент Крищенко Н.С.

2016

Содержание

[ВВЕДЕНИЕ 3](#_Toc366715862)

[1 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ](#_Toc366715863) 4

[1.1 Анализ предметной области](#_Toc366715864) 4

[1.2 Описание ограничений](#_Toc366715865) 6

[2 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ](#_Toc366715867) 10

3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 18

[ВЫВОДЫ](#_Toc366715868) 21

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc366715869)

# ВВЕДЕНИЕ

Отношение пользователей к web-приложениям на сегодняшний день остается неоднозначным. Достоинства браузерных приложений очевидны.

 Во-первых, пользователь не нуждается в установке на свою машину тяжеловесного программного обеспечения. Все, что требуется для полноценной работы - это браузер, обычно поставляемый вместе с операционной системой, и доступ в Интернет.

 Во-вторых, устанавливая приложения на свой компьютер, поневоле приходится брать на себя обязанности администратора, что доставляет неопытным пользователям массу хлопот. Приложение нужно установить и запустить, потом настроить под себя, а потом ни с того ни с сего вдруг возникают непонятные ошибки, требующие немедленного решения. В случае с браузерным приложением, фактически лежащем на сервере, беспокоиться об этом не придется.

В третьих, web-приложения не требовательны к ресурсам и не предъявляют никаких требований к аппаратной платформе. Это значит, что нет никакой разницы, сколько мегабайт оперативной памяти установлено на компьютере пользователя и из под какой операционной системы он работает. Лишь бы был браузер и доступ в Интернет, все остальное не так уж и важно.

И наконец, web-приложения позволяют своим пользователям быть по-настоящему мобильными. По сути, вы можете работать в сети, сохранять результаты своей работы на сервере и, в случае необходимости, иметь к ним доступ отовсюду.

К сожалению у web-приложений существуют и слабые стороны, которые, конечно не затмевают достоинств. Но последние на их фоне выглядят уже не так привлекательно.

 Для начала, интернет сейчас доступен, увы, не везде - по крайней мере в нашей стране. И во многих районах нашей "необъятной" стоимость трафика и ширина интернет-канал оставляют желать лучшего.

 Кроме того, существует огромное количество приложений, которые не могут быть заменены браузерными (по крайней мере в ближайшем будущем). Например, невозможно в браузере создавать сложные трехмерные модели.

 И наконец самый главный минус web-приложений -  многих пользователей смущает тот факт, что их данные будут храниться и обрабатываться где-то на чужом сервере. Ведь потенциально это может привести к утечке, потере или искажению информации (и в отдельных случаях наверняка приведёт). Не каждый рискнет выложить личную информацию в сеть.

В данном проекте реализовано веб-приложение «Информационная система «Биатлон»», которое будет очень полезно для любителей этого захватывающего зимнего вида спорта.

# ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

База данных «Биатлон» является основой для построения тематического веб-приложения. Иными словами она служит информационным источником для соответствующего проекта.

**1.1 Анализ предметной области**

“Биатлон” – информационная компьютерная система, предназначенная для хранения и управления данными о биатлоне.

Система предоставляет доступ к следующим данным:

* Информация о биатлонистах, которые выступают на международных соревнованиях и их результатах.
* Информация о дисциплинах, которые проходят на отдельных этапах Кубка мира.
* Информация о зачетах Кубка мира.

Целью системы является ознакомление со списком действующих спортсменов и получении информации о их успехах.

При анализе предметной области “Биатлон” были выделены следующие сущности:

* Биатлонист
* Дисциплина
* Этап Кубка мира
* Гонка
* Результат
* Общий зачет
* Медальный зачет

Данные сущности находятся в следующих связях друг с другом: биатлонист выступает на международных соревнованиях Кубка мира, которых состоит с определённого количества этапов. На каждом этапе спортсмен проходит приблизительно 3 гонки из существующих 6 дисциплин. В каждой в зависимости от результата, т.е. занятого биатлонистом места спортсмен получает некоторое количество баллов, которые идут в общий зачет, от которого зависит рейтинг биатлета. За попадание в тройку лучших биатлонист получает медаль, эта информация заносится в медальный зачет.

Логическая модель базы данных приведена на рисунке (рисунок 1.1).

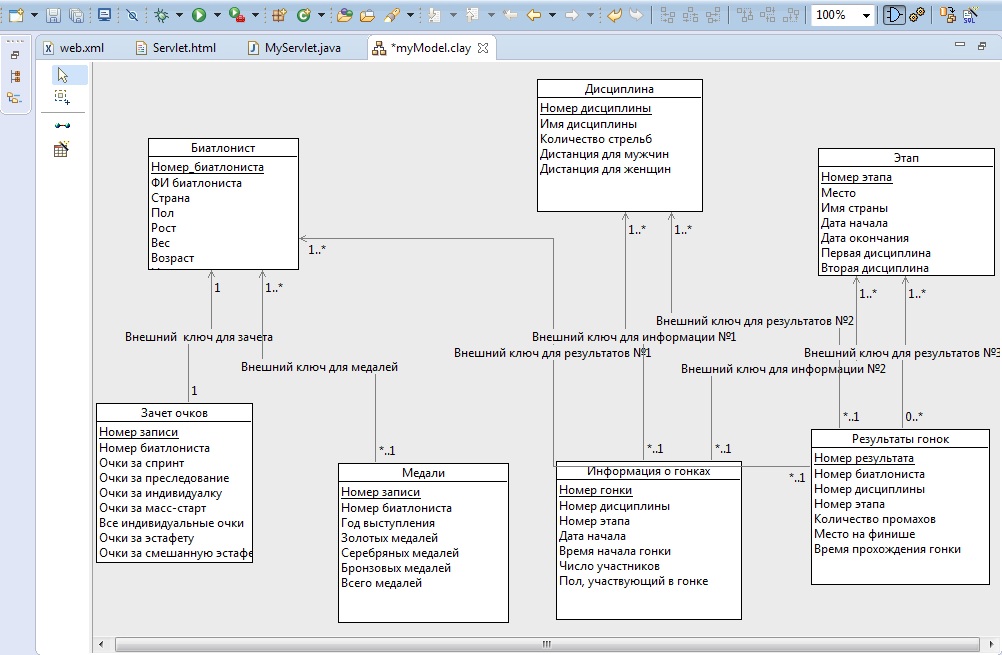


Рисунок 1.1 — Логическая модель базы данных

Физическая модель базы данных приведена на рисунке (рисунок 1.2).

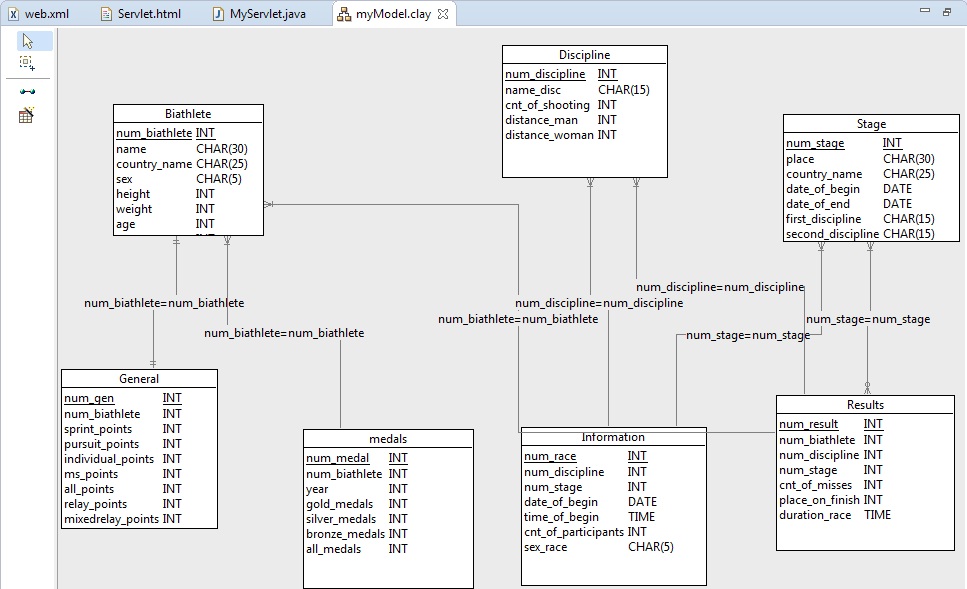


Рисунок 1.2 — Физическая модель базы данных

**1.2 Описание ограничений**

Далее произведем описание ограничений целостности, накладываемых на данные. Данное описание приведено в таблицах 1.1-1.7.

Таблица 1.1 — Ограничения целостности, накладываемые на данные

для биатлета

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название таблицы | Поле | Описание | Тип данных | Ограничение |
| Биатлет | num\_biathlete | Код биатлета | Serial | Первичный ключ, обязательность ввода, уникальность кода |
| name | Имя и фамилия | Char(30) | Обязательность ввода, уникальность |
| country\_name | Страна рождения | Char(25) | Обязательность ввода |
| sex | Пол | Char(5) | Обязательность ввода, проверка |
| height | Рост | integer | Обязательность ввода |
| weight | Вес | integer | Обязательность ввода |
| age | Возраст | integer | Обязательность ввода |

Таблица 1.2 — Ограничения целостности, накладываемые на данные

для дисциплины

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название таблицы | Поле | Описание | Тип данных | Ограничение |
| Дисциплина | num\_discipline | Код дисциплины | Serial | Первичный ключ, обязательность ввода |
| name\_disc | Название дисциплины | Char(20) | Обязательность ввода, уникальность |
| cnt\_of\_shooting | Количество огневых рубежей | integer | Обязательность ввода |
| distance\_man | Дистанция для мужчин | real | Обязательность ввода |
| distance\_woman | Дистанция для женщин | real | Обязательность ввода |

Таблица 1.3 — Ограничения целостности, накладываемые на данные

для этапов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название таблицы | Поле | Описание | Тип данных | Ограничение |
| Этап | num\_stage | Код этапа | Serial | Первичный ключ, обязательность ввода |
| place | Место проведения | Char(20) | Обязательность ввода |
| country\_name | Страна проведения | Char(25) | Обязательность ввода |
| begin\_date | Начало этапа | Date | Обязательность ввода |
| end\_date | Конец этапа | Date | Обязательность ввода |
| first\_disc | Первая дисциплина | Char(15) | Обязательность ввода |
| second\_disc | Вторая дисциплина | Char(15) | Обязательность ввода |
| third\_disc | Третяя дисциплина | Char(15) | Обязательность ввода |

Таблица 1.4 — Ограничения целостности, накладываемые на данные

для гонок

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название балицы | Поле | Описание | Тип данных | Ограничение |
| Гонка | num\_race | Код гонки | Serial | Первичный ключ, обязательность ввода |
| num\_discipline | Код дисциплины | Serial | Обязательность ввода,внешн ключ |
| num\_stage | Код этапа | Serial | Обязательность ввода,внешн ключ |
| date\_of\_begin | Дата начала | DATE | Обязательность ввода |
| time\_of\_begin | Время старта | time without time zone | Обязательность ввода |
| cnt\_of\_participants | Кол-во участников | integer | Обязательность ввода |
| sex\_race | Выступление мужчин/женщин | Char(5) | Обязательность ввода |

Таблица 1.5 — Ограничения целостности, накладываемые на данные

для результатов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название таблицы | Поле | Описание | Тип данных | Ограничение |
| Результат | num\_result | Код результата | Serial | Первичный ключ, обязательность ввода |
| num\_biathlete | Код биатлета | Serial | Обязательность ввода,внешн ключ |
| num\_discipline | Код дисциплины | Serial | Обязательность ввода,внешн ключ |
| num\_stage | Код этапа | Serial | Обязательность ввода,внешн ключ |
| cnt\_of\_misses | Кол-во промахов | integer | Обязательность ввода |
| place\_on\_finish | Место на финише | integer | Обязательность ввода |
| duration\_passing\_of\_race | Время прохождения трассы | time without time zone | Обязательность ввода |

Таблица 1.6 — Ограничения целостности, накладываемые на данные

для общего зачета

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название таблицы | Поле | Описание | Тип данных | Ограничение |
| Общий зачет | num\_gen | Код общего зачета | Serial | Первичный ключ, обязательность ввода |
| num\_biathlete | Код биатлета | Serial | Обязательность ввода, внешний ключ |
| sprint\_points | Очки за спринт | integer | Обязательность ввода |
| pursuit\_points | Очки за гонку пресл. | integer | Обязательность ввода |
| ms\_points | Очки за масстарт | integer | Обязательность ввода |
| individual\_points | Очки за индив.гонку | integer | Обязательность ввода |
| All\_points | Все очки | integer | Обязательность ввода, проверка |
| Relay\_points | Очки за эстафету | integer | Обязательность ввода |
| Mixed\_points | Очки за смеш.эстафету | integer | Обязательность ввода |

Таблица 1.7 — Ограничения целостности, накладываемые на данные

для медального зачета

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название таблицы | Поле | Описание | Тип данных | Ограничение |
| Медальный зачет | num\_medal | Код медального зачета | Serial | Первичный ключ, обязательность ввода |
| num\_biathlete | Код биатлета | Serial | Обязательность ввода,внешний ключ |
| year | Год добытия медали | Integer | Обязательность ввода |
| gold\_medals | Количество золотых медалей | Integer | Обязательность ввода |
| silver\_medals | Количество серебреных медалей | Integer | Обязательность ввода |
| bronze\_medals | Количество бронзовых медалей | Integer | Обязательность ввода |
| all\_medals | Общее количество медалей | Integer | Обязательность ввода, проверка |

**2 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ**

Веб-приложение «Информационная система «Биатлон»» - веб-приложение, сделанное на основе ранее полученных знаний с учебных предметов «Организация баз данных», «Прикладное программирование», «Проектирование веб-интерфейсов».

Разработанная база данных уже была описана ранее в пункте 1.

Представленное веб-приложение было сделано сделано на платформе Eclipse. Был выбран язык программирования Java, на котором были в свою очередь реализованы классы для работы с веб-приложением. Были использованы ранее получены умения при использовании технологий Servlet, Java Server Pages, Java Persistance Annotations. Для связи с базой данных «Биатлон», которая была сделана в среде PostgreSQL, был использован драйвер JDBC. Также были подключены нужные библиотеки для коректной работы приложения.

На рисунке 2.1 предствлена диаграмма классов.

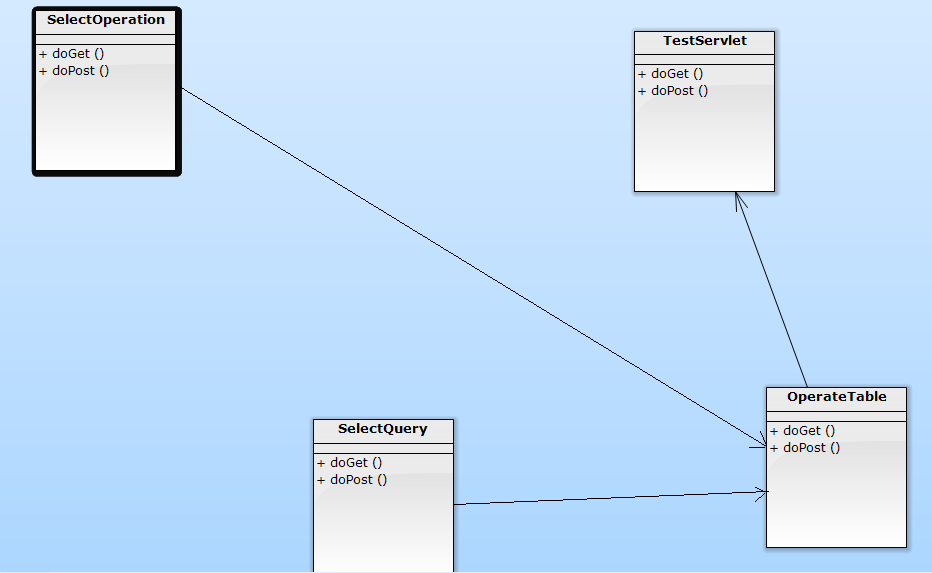


Рисунок 2.1 – Диаграмма классов веб-приложения «Информационная система «Биатлон»»

Далее представлен листинг некоторых классов, созданных для данного веб-приложения.

**Текст класса для тестирования сервлета TestServlet:**

package web\_view;

import java.io.IOException;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

import javax.persistence.Query;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import controller.JPAController;

import model.Biathlete;

/\*\*

\* Servlet implementation class TestServlet

\*/

@WebServlet("/TestServlet")

public class TestServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public JPAController controller = new JPAController();

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public TestServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

EntityManagerFactory emf = Persistence.createEntityManagerFactory("SupportProject");

EntityManager em = emf.createEntityManager();

Query q = em.createQuery("select x from Biathlete x order by x.id");

List<Biathlete> list = q.getResultList();

for (Biathlete s : list) {

System.out.println(s);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**Текст класса для выбора операции действия над данными SelectOperation:**

package web\_view;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class SelectOperation

\*/

@WebServlet("/SelectOperation")

public class SelectOperation extends HttpServlet {

private static final long serialVersionUID = 1L;

public static int operation;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public SelectOperation() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

operation = Integer.parseInt(request.getParameter("operation"));

String tableClassName = SelectTable.tableName;

String formName = "";

if(operation == 4){

formName = "getId.html";

}else if(tableClassName.equals("model.Biathlete")){

if(operation == 1){

formName = "inputbiathlete.html";

}else if(operation == 2){

formName = "editbiathlete.html";

}

}else if(tableClassName.equals("model.Race")){

if(operation == 1){

formName = "inputrace.html";

}else if(operation == 2){

formName = "editrace.html";

}

}else if(tableClassName.equals("model.Results")){

if(operation == 1){

formName = "inputresults.html";

}else if(operation == 2){

formName = "editresults.html";

}

}else if(tableClassName.equals("model.General")){

if(operation == 1){

formName = "inputgeneral.html";

}else if(operation == 2){

formName = "editgeneral.html";

}

}else if(tableClassName.equals("model.Medals")){

if(operation == 1){

formName = "inputmedals.html";

}else if(operation == 2){

formName = "editmedals.html";

}

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**Текст класса для выполнения операций над данными таблицы OperationTable:**

package web\_view;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import model.DataBaseModel;

/\*\*

\* Servlet implementation class OperateTable

\*/

@WebServlet("/OperateTable")

public class OperateTable extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public OperateTable() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

private DataBaseModel findObject(Class clazz, int id){

return null;

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

**Текст файла showTable.jsp**

<%@ page import="javax.swing.table.TableModel" %>

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>showTable</title>

</head>

<body>

<%

String tableName = (String)request.getAttribute("tableName");

TableModel model = (TableModel) request.getAttribute("model");

%>

<center>

<b> Таблиця <%=tableName%></b>

<table border="1">

<tr>

<%

int nCol = model.getColumnCount();

for(int i = 0; i < nCol; i++){

String h = model.getColumnName(i);

%>

<th width="100%"><%=h%></th>

<% } %>

</tr>

<%

int nRow = model.getRowCount();

for(int r = 0; r < nRow; r++){

%>

<tr>

<%

for(int j = 0; j <nCol; j++){

String str = model.getValueAt(r, j).toString();

%>

<td width="100" align="center"> <%=str%> </td>

<% } %>

</tr>

<% } %>

</table>

</center>

</body>

</html>

Со стороны веб-интерфейсов были сделаны html-страницы для оформления самой страницы веб-приложения.

**Листинг index.html:**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Біатлон</title>

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-theme.min.css">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>

<style>

body{

background-color: #white; /\* Цвет фона веб-страницы \*/

}

</style>

</head>

</head>

<body>

<center>

<h1><p> Інформаційна система "Біатлон"</p></h1>

<p><img src="michen.JPG" width="1205" height="200" alt = "biathlon" ></p>

<iframe name="forMenu" src="menuTable.html" width="405" height="400" align = top></iframe>

<iframe name="forOperation" src="generalTable.html" width="800" height="400"> </iframe>

<p><img src="stadium.JPG" width="1205" height="300" alt = "biathlon" ></p>

</center>

</body>

</html>

**Листинг menuTable.html:**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-theme.min.css">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>

<style>

</style>

</head>

</head>

<body>

<center-left>

<form action="SelectTable" target="forTable" name=".SelectTable"

method="POST">

<table width="80">

<p>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Інформація про біатлета"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Інформація про дисципліни"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Інформація про етапи"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Інформація про гонки"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Результати"></td>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Загальний залік біатлетів"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Медальний залік біатлетів"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Додати дані"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Видалити дані"></td>

</tr>

<tr>

<td align="left"><input button class="btn btn-info" type="submit" style="width:400Px" name="tableName"

value="Змінити дані"></td>

</tr>

</p>

</table>

</form>

<br>

</center-left>

</body>

</html>

**3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

В этом разделе представлены функции работы с веб-приложением «Информационная система «Биатлон»».

На рисунке 3.1 представлена главная страница веб-приложения «Информационная система «Биатлон»».

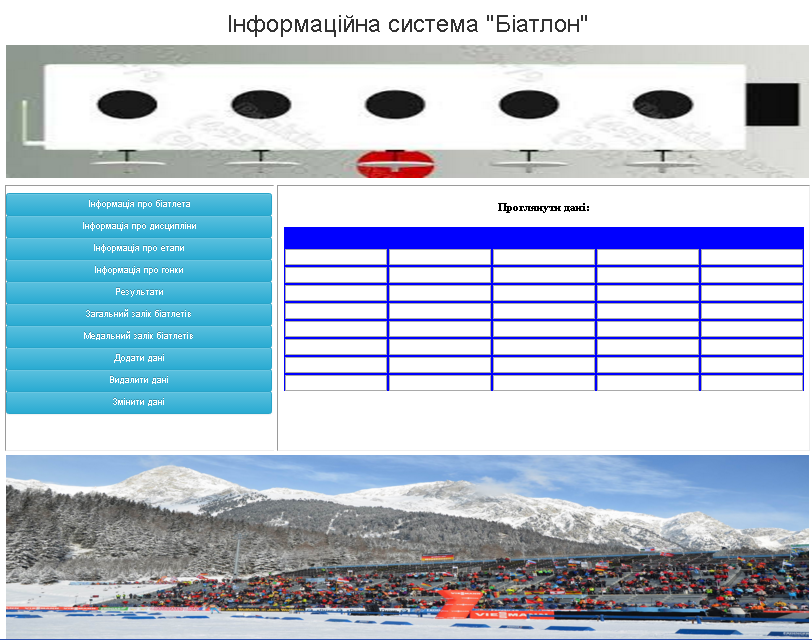
****

Рисунок 3.1 – Главная страница веб-приложения «Информационная система «Биатлон»»

Как видно с рисунка 3.1 главная страница веб-приложения состоит из заголовка, 2х картинок вверху и внизу страницы, а также, что является основным, – это кнопки для вызова нужной информации с базы данных, которые формируют рядом стоящею таблицу, с которой мы можем получить 7 основополагающих пунктов информации – о биатлонистах, дисциплинах, этапах, гонках, результатах, общем и медальном зачетах. Также разрешается отредактировать при вызове соответствующих форм данные – можно добавить, удалить или изменить определённую информацию.

Далее рассмотрим основные возможности веб-приложения, используя примеры окон.

На рисунке 3.2 изображена информация в таблице, которая заполняется данными с базы данных при нажатии на кнопку «Інформація про біатлета».



Рисунок 3.2 – Окно «Інформація про біатлета»

На рисунке 3.3 изображена информация в таблице, которая заполняется данными с базы данных при нажатии на кнопку «Інформація про дисципліни».



Рисунок 3.3 – Окно «Інформація про дисципліни»

На рисунке 3.4 изображена информация в таблице, которая заполняется данными с базы данных при нажатии на кнопку «Інформація про етапи».



Рисунок 3.4 – Окно «Інформація про етапи»

На рисунке 3.5 изображена информация в таблице, которая заполняется данными с базы данных при выполнении запроса БД при нажатии на кнопку «Загальний залік».



Рисунок 3.5 – Окно «Загальний залік»

На рисунке 3.6 изображена информация в таблице, которая заполняется данными с базы данных при выполнении запроса БД при нажатии на кнопку «Медальний залік».



Рисунок 3.6 – Окно «Медальний залік»

На рисунке 3.7 представлен пример формы добавления биатлета. Также данные о биатлетах и о других таблицах можно удалять и изменять.

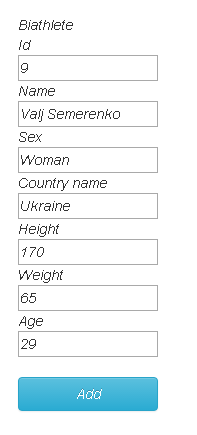


Рисунок 3.7 – Добавления биатлета

# выводы

Таким образом, разработка веб-приложений предоставляет широкие возможности по созданию для компаний многофункциональных онлайн-инструментов для оптимизации или решения различных бизнес-задач.

В данной учебной практике было изучено основы проектирования веб-приложений. Было создано веб-приложение «Информационная система «Биатлон»». При выполнении работы возникали небольшие трудности с определёнными процессами, так как это делалось впервые и часто первый блин выходит комом. Эта система может бать очень полезной и информационной как для профессиональных любителей этого зимного вида спорта, так и для новичков.

Работа была умеренной сложности, но очень интересной и увлекательной. Теперь имея представление, что такое веб-приложения, можно развивать свои умения и иметь цель достичь определённых успехов в этой сфере.

# Список использованых источников

1. Уроки по созданию баз данных на сайте <http://www.lessons-tva.info/edu/e-inf2/m2t4_3.html>;
2. Видеоуроки создания веб-приложений на <http://www.youtube.com>;
3. Java технології прикладного програмування. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Java та C# технології прикладного програмування» для студентів напрямів підготовки6.050102 – “Комп’ютерна інженерія” 6.050103 – “Програмна інженерія”, частина перша. /Укл.: Бивойно П.Г., Бивойно Т.П. – Чернігів: ЧНТУ, 2016. – 82 с.;
4. Технології створення веб-сайтів. Методичні вказівки до лабораторних робіт з вибіркової дисципліни «Проектування WEB-інтерфейсів» для студентів напряму підготовки 6.050102 «Комп’ютерна інженерія». – Чернігів: ЧДТУ, 2013. - с., рос. мовою.