# Введение

Интернет сегодня занимает важную роль в нашей повседневной жизни. Существует большое количество вещей, которые можно сделать с помощью Интернета. Можно сказать, что с развитием Интернета мы прогрессируем в каждой сфере жизни, поскольку интернет не только упрощает решение многих задач, но и экономит массу времени. Сегодня Интернет используется для самых различных целей, из основных можно выделить:

1. Связь

 На данный момент самое простое, что можно делать при помощи Интернета, - это общаться с людьми, живущими далеко от нас, с чрезвычайной легкостью. Раньше такое общение было сложной задачей, а теперь человек может легко связаться с близкими, которые находятся в другой части мира. Электронная почта, социальные сети плотно вошли в нашу жизнь и облегчили ее в значительной степени.

2. Образование

 Образование - это одна из лучших вещей, которую может предоставить нам интернет. Есть ряд книг, справочников, онлайновых справочных центров и других учебных материалов, ориентированных на изучение в интернете, которые могут сделать процесс обучения более легким. В любое время вы можете посетить нужные вам Интернет-ресурсы и получить ту информацию в которой нуждаетесь.

3. Финансы

Благодаря интернету, теперь вам не нужно стоять в очереди вашего банка, чтобы выполнить какую-либо транзакцию. Теперь можно легко выполнять финансовые транзакции через Интернет, а также можно можете легко купить или продать что-либо не выходя из дома.

В окна браузеров перешли многие приложения, которые раньше требовали наличия мощного компьютера. Сегодня же их можно легко использовать имея под рукой смартфон, или не самый мощный компьютер.

***Цель дипломной работы*** – создать веб приложение с использованием языков PHP, HTML, JS, CSS и базы данных MYSQL.

***Задачи дипломной работы:***

1. изучение теоретических основ разработки веб приложений.
2. выбор платформы и, соответственно, среды программирования, на которой будет реализовано приложение;
3. реализация приложения.

***Объектом исследования*** в данной работе являются веб-приложение

***Предмет исследования:*** процесс разработки веб-приложения

В качестве языков реализации приложения выбраны php, html, css, java script, в качестве базы данных- mysql.

Работа состоит из трех глав. В первой главе приводится описание операционных систем мобильных устройств. Во второй главе рассматриваются технологии создания мобильных приложений. Глава 3 посвящена разработке мобильного приложения «Расписание общественного транспорта города Могилева».

Дипломная работа содержит && страниц, && рисунков, 1приложение, & использованных литературных источников.

# Описание предметной области

## Концепция веб-приложения «органайзер-преподавателя»

За последние несколько лет стала очень актуальна разработка веб-приложений. Человеческая жизнь всё больше переходит на просторы интернета. Сегодня мы имеем возможность хранить и обрабатывать большие объёмы информации на удалённых компьютерах, называемых серверами, посредством обычного веб-браузера так как это удобно с точки зрения доступности и достаточно надёжно.

Преподавателям приходится собирать, хранить и обрабатывать массу информации на своей работе. К такой информации можно отнести состав учебных групп, информацию об учащихся, расписании занятий и иных мероприятий на которых нужно будет присутствовать, хранить и иметь быстрый доступ к различным материалам, необходимым для обучения, приходится подсчитывать различные данные по результатам своей работы такие как средний балл и посещаемость. Для того, чтобы хоть немного упростить эти задачи было решено разработать веб-приложение, выполняющее аналогичные задачи.

Объектом разработки является приложение, которое позволит собрать необходимую информацию, сгруппировать её таким образом, чтобы пользователь в несколько кликов мог получить интересующую информацию и результат её анализа. Это позволит пользователю быстрее ориентироваться и сэкономит много времени.

## Технические особенности веб-приложений

Существенное преимущество веб-приложений заключается в том, что их функции должны выполняться независимо от операционной системы пользователя приложения. Вместо того, чтобы писать различные версии для Microsoft Windows, Mac OS X, GNU/Linux и других операционных систем, приложение создаётся один раз для произвольно выбранной платформы и на ней разворачивается. Однако различная реализация HTML, CSS, DOM и других спецификаций в браузерах может вызвать проблемы при разработке веб-приложений и последующей поддержке. Кроме того, возможность пользователя настраивать многие параметры браузера, например, размер шрифта, цвета, отключение поддержки сценариев, может препятствовать корректной работе приложения.

Другой, менее универсальный, подход заключается в использовании Adobe Flash, Silverlight или Java-апплетов для полной или частичной реализации пользовательского интерфейса. Поскольку большинство браузеров, как правило, с помощью плагинов, поддерживает эти технологии. Так как они предоставляют программисту больший контроль над интерфейсом, они способны обходить многие несовместимости в конфигурациях браузеров, хотя несовместимость между Java- или Flash-реализациями на стороне клиента также может приводить к различным осложнениям.

Однако такой подход не очень популярен, в силу безопасности и уже с 2015 года технологию Adobe Flash и java-апплеты не поддерживают Chrome, Safari, и другие популярные браузеры.

## Паттерн MVC

Создание и модификация веб-приложения представляет собой достаточно сложную и трудоёмкую задачу, над которой, как правило трудится не один, а целая команда специалистов, каждый из которых выполняет поставленную перед ним задачу. Их эффективное взаимодействие не было бы возможным без некоторых приёмов. К одному из таких приёмов можно отнести паттерн  **MVC.**  Открытый в конце 1970-х, он представляет собой шаблон проектирования архитектуры программного обеспечения, основной задачей которого является отделение функций работы с данными от их представления. Теоретически, грамотно спроектированное MVC-приложение позволяет фронтенд и бэкенд разработчикам в ходе работы не вмешиваться в зоны ответственности друг друга.

Название паттерна происходит от аббревиатуры трёх слов: Model (модель), View (представление) и Controller (контроллер).

Модель используется для доступа и манипулирования данными. В большинстве случаев модель — это то, что используется для доступа к хранилищу данных (например, базе данных). Модель предоставляет интерфейс для поиска данных, их создания, модификации и удаления из хранилища. В контексте паттерна MVC модель является посредником между представлением и контроллером.

Крайне важной чертой модели является то, что технически она не имеет никаких знаний ни о том, что происходит с данными в контроллере и представлении. Модель никогда не должна делать или ожидать каких-либо запросов в других компонентов паттерна.

Представление

Представление - это то, где данные, полученные от модели, выводятся в нужном виде. В традиционных веб-приложениях, разработанных в рамках MVC-паттерна, представление - это часть системы, где выполняется генерация HTML-кода. Представление также отвечает за получение действий от пользователя с тем чтобы отправить их контроллеру. Например, представление отображает кнопку в пользовательском интерфейсе, а после её нажатия вызывает соответствующее действие контроллера.

Контроллер

Контроллер — это последняя часть связки MVC. Задачей контроллера является получение данных от пользователя и манипуляция моделью. Именно контроллер, и только он, является той частью системы, которая взаимодействует с пользователем.

В двух словах контроллер можно описать как сборщик информации, передающий её модели для обработки и хранения. Он не должен делать ничего с данными, а только лишь уметь получать их от пользователя.

# Выбор и описание базовых технологий для создания веб-приложений

## Архитектура клиент-сервер

Как правило компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами, другие имеют возможность обращаться к этим ресурсам. Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса. Клиент и сервер какого-либо ресурса могут находится как на одном компьютере, так и на различных компьютерах, связанных сетью.

Любое веб-приложение построено на клиент-серверной архитектуре, где клиентом выступает браузер, а сервером удалённый компьютер, который отвечает на результаты запроса браузера, предоставляя ему необходимый контент, при необходимости обращаясь к базам данных.