

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук  
Кафедра информационных систем

Система учёта успеваемости и посещаемости  
студентов факультета «Зачётка.ру»

Курсовая работа

*09.03.02 Информационные системы и технологии*  
*Информационные системы и сетевые технологии*  
*6 семестр 2022/2023 учебного года*

|               |       |                                     |
|---------------|-------|-------------------------------------|
| Зав. кафедрой | _____ | к. т. н., доцент Д.Н. Борисов       |
| Обучающийся   | _____ | ст. 3 курса оч. отд. А.О. Левин     |
| Обучающийся   | _____ | ст. 3 курса оч. отд. Ф.Г. Парамонов |
| Обучающийся   | _____ | ст. 3 курса оч. отд. Д.Д. Боброва   |
| Руководитель  | _____ | ст. преп. В.С. Тарасов              |

Воронеж 2023

## Содержание

|  |    |
|--|----|
| Введение .....   | 3  |
| 1 Постановка задачи.....                                 | 4  |
| 2 Цель создания программы.....                           | 4  |
| 3 Анализ предметной области .....                        | 5  |
| 3.1 Сфера применения .....                               | 5  |
| 3.2 Технический обзор .....                              | 6  |
| 3.3 Анализ аналогов .....                                | 8  |
| 3.4 Требования к функциональности .....                  | 11 |
| 3.4.1 Неавторизованный пользователь.....                 | 11 |
| 3.4.2 Авторизованный пользователь .....                  | 12 |
| 3.4.3 Администратор .....                                | 12 |
| 3.4.4 Преподаватель .....                                | 12 |
| 3.4.5 Студент.....                                       | 14 |
| 3.5 Требования, не касающиеся функциональной части ..... | 15 |

## **Введение**

В настоящее время использование информационных технологий в образовательном процессе является неотъемлемой частью современного образования. Технологии помогают не только автоматизировать рутинные задачи, но и упростить и улучшить процессы обучения и оценки результатов учебной деятельности.

Одним из важных аспектов образовательного процесса является учет успеваемости студентов, который позволяет оценить их академические результаты и прогресс в учебе. В традиционных методах учета успеваемости используется ручное ведение журналов, что может быть неэффективным и неудобным для использования. Кроме того, ручное ведение журналов не позволяет проводить детальный анализ результатов учебной деятельности студентов, что может затруднять принятие обоснованных решений.

В связи с этим возникает необходимость в использовании специализированных программных продуктов, которые позволяют автоматизировать процесс учета успеваемости студентов. В настоящее время существует множество систем учета успеваемости студентов, в том числе веб-приложений, которые позволяют не только автоматизировать процесс учета, но и проводить анализ результатов учебной деятельности.

## **1 Постановка задачи**

В ходе данной работы нужно разработать веб-приложение для учёта успеваемости и посещаемости студентов. Для этого необходимо:

- разработать функциональные требования к веб-приложению, учитывающие потребности пользователей и особенности учета успеваемости и посещаемости студентов;
- спроектировать базу данных для хранения информации об успеваемости и посещаемости студентов, обеспечивая ее надежность и масштабируемость;
- реализовать веб-приложение с необходимым функционалом для выставления оценок и посещаемости, анализа результатов учебной деятельности студентов;
- разработать интерфейс пользователя, удобный для работы преподавателей и студентов;
- протестировать веб-приложение на соответствие требованиям, выявить и устранить возможные ошибки;
- обеспечить безопасность данных и защиту от несанкционированного доступа;

Выполнение этих задач позволит создать веб-приложение, которое будет эффективно выполнять свои функции, обеспечивать удобство использования и защиту данных.

## **2 Цель создания программы**

Цель данной курсовой работы – создание веб-приложения для последующей интеграции его в учебные заведения. Веб-приложение должно обеспечить возможность учета преподавателями и просмотра студентами успеваемости и посещаемости занятий на факультете учебного заведения. Оно позволит автоматизировать процесс ведения журналов успеваемости и посещаемости, а также анализа результатов учебной деятельности студентов.

Данным приложением будет удобно пользоваться как студентам, так и преподавателям: для студентов вся необходимая информация собрана в одном месте, а для преподавателей предусмотрена функция автоматической отметки посещаемости студентов на занятии.

В рамках работы будет рассмотрено проектирование и реализация функциональных возможностей веб-приложения, в том числе разработка базы данных для хранения информации об успеваемости студентов, разработка интерфейса пользователя и функционала для анализа результатов.

### **3 Анализ предметной области**

#### **3.1 Сфера применения**

Такие приложения могут быть использованы в различных учебных заведениях – от школ до вузов. Они предназначены для автоматизации учета успеваемости студентов и облегчения работы преподавателей и администрации образовательных учреждений.

Программное обеспечение для учета успеваемости может быть использовано для следующих задач:

- учет посещаемости студентов: с помощью приложения преподаватели могут отмечать, кто был на занятии, а кто - нет. Также это помогает администрации учебного заведения контролировать посещаемость студентов и отслеживать их активность;
- учет оценок: приложение позволяет преподавателям вводить оценки и оценивать работы студентов. Оценки могут быть записаны в журнал, а также использоваться для создания отчетов и анализа успеваемости;
- учёт посещаемости: приложение позволяет преподавателям отмечать, посещал ли студент занятия. Это поможет преподавателю принимать решения по поводу оценки работы

студента в течение семестра при возникновении спорных ситуаций;

- анализ успеваемости студентов: приложение может собирать данные о успеваемости студентов и анализировать их. Это позволяет выявлять проблемные моменты и проводить анализ эффективности обучения;

Таким образом, веб-приложения для учета успеваемости студентов могут быть использованы в различных образовательных учреждениях для автоматизации процессов учета и анализа успеваемости студентов, что в свою очередь способствует повышению качества обучения и улучшению работы преподавателей.

### **3.2 Технический обзор**

Рассмотрим технические аспекты, которые следует учитывать при создании веб-приложения для учета успеваемости студентов.

Язык программирования – необходимый инструмент для создания любого веб-приложения. При создании приложения для учета успеваемости студентов можно использовать различные языки программирования. Так, PHP является наиболее популярным языком программирования для создания веб-приложений. Он широко используется для разработки сайтов и веб-приложений, основанных на системе управления контентом (Content management system, CMS). Python также популярен среди разработчиков веб-приложений и обладает мощными инструментами для создания сложных приложений. Ruby, Java и другие языки программирования также могут использоваться для создания веб-приложений.

Также фреймворки являются важными инструментами для создания веб-приложений, поскольку они позволяют разработчикам быстро и эффективно создавать приложения, используя готовые компоненты. Существует множество фреймворков для создания веб-приложений, включая Laravel, Django, Ruby on Rails, Spring и другие. Laravel и Django являются

популярными фреймворками, которые обладают мощными инструментами для создания веб-приложений и позволяют разработчикам быстро и эффективно создавать приложения.

Базы данных необходимы для хранения данных, связанных с учетом успеваемости студентов. Для создания веб-приложения можно использовать такие базы данных, как MySQL, PostgreSQL, MongoDB или другие. MySQL является наиболее распространенной базой данных, используемой в веб-разработке, и обладает хорошей производительностью и широким набором инструментов для управления данными. PostgreSQL также популярен среди разработчиков веб-приложений и обладает мощными функциями для управления данными. MongoDB является NoSQL базой данных, которая позволяет хранить данные в формате JSON.

Интерфейс – важная часть любого веб-приложения. Для создания интерфейса пользователя можно использовать различные технологии, включая HTML, CSS и JavaScript. HTML является языком разметки, который определяет структуру и содержимое веб-страницы. CSS позволяет определять внешний вид и стиль веб-страницы, включая цвета, шрифты и расположение элементов. JavaScript позволяет добавлять динамическое поведение к веб-странице, например, валидацию форм, анимацию и интерактивность.

Системы контроля версий позволяют разработчикам работать совместно над кодом, отслеживать изменения и вносить правки безопасно. Для управления кодом при создании веб-приложения для учета успеваемости студентов можно использовать системы контроля версий, такие как Git.

Для размещения веб-приложения можно использовать различные хостинг-провайдеры, включая Timeweb, Spaceweb, Reg.ru и другие. Эти провайдеры предоставляют облачные ресурсы для размещения и масштабирования веб-приложений.

Безопасность является важным аспектом при создании веб-приложений для учета успеваемости студентов. Веб-приложения могут быть подвержены атакам, таким как SQL-инъекции, кросс-сайтовые скрипты и другие. Для

обеспечения безопасности приложения можно использовать различные методы, включая хэширование паролей, проверку ввода данных и использование SSL-сертификатов.

Масштабируемость также является важным аспектом для веб-приложений, которые должны обслуживать большое количество пользователей. Для обеспечения масштабируемости можно использовать различные технологии, включая горизонтальное масштабирование и вертикальное масштабирование.

Таким образом, создание веб-приложения для учета успеваемости студентов подразумевает под собой использование различных технологий, включая языки программирования, фреймворки, базы данных, интерфейс пользователя, системы контроля версий, хостинг, безопасность и масштабируемость.

### **3.3 Анализ аналогов**

На рынке существует множество различных веб-приложений для учета успеваемости студентов. Рассмотрим некоторые из наиболее популярных аналогов, а также их преимущества и недостатки:

Moodle — это популярная система управления обучением с открытым исходным кодом. Она предоставляет функциональность для учета посещаемости, выставления оценок и анализа успеваемости. Moodle также имеет множество дополнительных модулей, которые расширяют ее функциональность.

Преимущества:

- бесплатный и имеет открытый исходный код;
- множество дополнительных модулей и плагинов для расширения функциональности;
- поддержка многих языков.

Недостатки:

- требуется некоторое время для изучения и освоения системы;



- управление и настройка может потребовать некоторых технических знаний;
- некоторые пользователи могут считать интерфейс системы неудобным или устаревшим.

Blackboard — это еще одна популярная система управления обучением, которая используется во многих учебных заведениях по всему миру. Она предоставляет множество функций, включая учет посещаемости, выставление оценок, обмен сообщениями и т.д.

Преимущества:

- удобный интерфейс и интуитивно понятный дизайн;
- множество интегрированных инструментов для управления обучением;
- поддержка многих языков.

Недостатки:

- высокая стоимость, особенно для небольших учебных заведений;
- некоторые функции могут быть ограничены или требовать дополнительной оплаты;
- некоторые пользователи могут считать систему слишком сложной для использования.

Gradescope — это веб-приложение, которое специализируется на проверке заданий и выставлении оценок. Оно позволяет преподавателям быстро проверять большие объемы работ и выставять оценки.

Преимущества:

- быстрое выставление оценок и проверка больших объемов работ;
- возможность загрузки и проверки различных форматов файлов;
- интеграция с другими системами управления обучением.

Недостатки:

- основная функциональность системы ограничена проверкой заданий и выставлением оценок;

- нет возможности учета посещаемости или анализа успеваемости студентов;
- стоимость использования может быть высокой для небольших учебных заведений.

Easyclass — это веб-приложение, которое предоставляет функциональность для учета посещаемости, выставления оценок и обмена сообщениями. Easyclass также позволяет преподавателям создавать онлайн-уроки и делиться материалами с учениками.

Преимущества:

- удобный и простой в использовании интерфейс;
- множество функций для учета посещаемости, выставления оценок и обмена сообщениями;
- поддержка многих языков.

Недостатки:

- некоторые функции могут быть ограничены или требовать дополнительной оплаты;
- не всегда достаточно гибок для учета специфических требований учебных заведений;
- могут возникать проблемы с безопасностью данных.

SchoolTime — это интегрированная платформа для управления учебным процессом, которая включает функциональность для учета посещаемости, выставления оценок, анализа успеваемости и т.д. Она также позволяет создавать расписания занятий и отправлять уведомления студентам и их родителям.

Преимущества:

- удобный и простой в использовании интерфейс;
- множество функций для учета посещаемости, выставления оценок и управления расписанием;
- модуль учета финансов и возможность онлайн-оплаты;
- поддержка многих языков.

Недостатки:

- некоторые функции могут быть ограничены или требовать дополнительной оплаты;
- нет возможности для интеграции с другими системами управления обучением;
- могут возникать проблемы с безопасностью данных.

Это лишь некоторые из популярных аналогов веб-приложений для учета успеваемости студентов. Несмотря на то, что каждый из них имеет свои преимущества и недостатки, веб-приложение для учета успеваемости студентов, созданное специально для конкретной учебной организации, может быть более гибким и удобным для использования, так как учитывает специфические потребности этой организации и не имеет ограничений, связанных с использованием сторонних систем.

### **3.4 Требования к функциональности**

Для реализации требуемых функций в веб-приложении предусмотрены три типа авторизованных пользователей: администратор, преподаватель, студент. Каждый тип будет иметь свой набор функциональных возможностей, а неавторизованные пользователи не смогут получить доступ к полному функционалу. Часть функционала, который присутствует у всех пользователей, для удобства расписана отдельно.

#### **3.4.1 Неавторизованный пользователь**

Неавторизованный пользователь имеет доступ только к странице авторизации. Она же «вход в систему».

На странице «вход в систему» представлены поля для ввода логина, пароля и кнопка «вход».

### **3.4.2 Авторизованный пользователь**

У любого авторизованного пользователя вверху каждой страницы представлена шапка, на которой есть кнопки «сменить пароль» и «выход».

При нажатии на кнопку «сменить пароль» пользователь попадает на страницу «смена пароля», на которой представлены три поля для заполнения:

- текущий пароль;
- новый пароль;
- повторный ввод нового пароля.

После заполнения всех полей пользователь может нажать кнопку «сменить» для смены пароля.

При нажатии на кнопку «выход», пользователь может выйти из своей учётной записи, после чего он попадёт на страницу авторизации.

### **3.4.3 Администратор**

После авторизации в системе администратор попадает на страницу «панель администратора», содержащую перечень доступных ему функций. На данной странице расположены кнопки «добавить администратора», «добавить студента», «добавить преподавателя», «добавить группу», «добавить пару», «добавить дисциплину», «удалить пару», «удалить дисциплину», «изменить дисциплину». При нажатии на любую из кнопок, откроется всплывающее окно с функционалом, соответствующим названию кнопки. Также на данной странице представлены кнопки «перевести на следующий семестр», «перевести на предыдущий семестр», позволяющие переключить текущий семестр для всех студентов.

### **3.4.4 Преподаватель**

После авторизации в системе преподаватель попадает на страницу «выбор дисциплины», отображающую все преподаваемые им дисциплины. По нажатию на наименование дисциплины происходит переход на страницу

«выбор группы», содержащую список групп студентов. Также на этой странице есть кнопка «проверка посещаемости» при нажатии на которую откроется всплывающее окно, в котором преподаватель выбирает дату и время занятия, а также длительность процесса проверки посещаемости. Возможная длительность процесса – от 10 секунд до 15 минут. После выбора параметров, преподаватель нажимает на кнопку «запустить», тем самым начиная процесс проверки посещаемости. После начала процесса, преподавателю необходимо сообщить находящимся на занятии студентам выданный ему системой код посещаемости – сгенерированную последовательность длиной в 7 символов, состоящую из цифр и букв английского алфавита. Также преподаватель может предоставить студентам для сканирования специальный QR-код, содержащий в себе ссылку, перейдя по которой авторизованный в системе студент, отмечает своё присутствие на занятии. После завершения процесса проверки посещаемости, будет отображен список групп студентов, которые отметились на занятии с указанием для каждой группы числа студентов, отметивших своё присутствие. Также необходимо отобразить общее число присутствующих на занятии студентов. Сразу после проведения проверки посещаемости, если преподаватель сомневается в том, что студенты отметились добросовестно, он может нажать на кнопку «сбросить учтённую сейчас посещаемость», тем самым аннулировав отмеченную студентами посещаемость по этому занятию.

Со страницы «выбор группы» по нажатию на номер группы можно перейти на страницу «учёт посещаемости по дисциплине», содержащую таблицу посещаемости студентов этой группы. В эту таблицу преподавателем вносятся данные о посещаемости студентов. Данными о посещаемости является информация о пропуске студентом конкретного занятия, общее количество пропусков, а также количество пропущенных студентом занятий, разделённое на следующие категории: «Н» - пропуск без уважительной причины, «П» - пропуск по уважительной причине, «Б» - пропуск по болезни. Выше таблицы есть пункт, отображающий общее количество занятий.

Со страницы «учёт посещаемости по дисциплине» возможен переход на страницу «учёт успеваемости по дисциплине», содержащую таблицу оценок студентов этой группы. В эту таблицу преподавателем вносятся данные об успеваемости студентов. Данными об успеваемости являются оценки за три контрольные точки (аттестации), средний балл за три контрольные точки (аттестации), оценка за экзамен, итоговая оценка. Оценка за каждую из аттестаций выставляется в виде балла от 0 до 50. Средний балл за аттестации рассчитывается системой автоматически в виде балла от 0 до 50. Оценка за экзамен является баллом от 0 до 100. Для дисциплин с типами отчётности «зачёт» и «зачёт с оценкой», оценка за экзамен не выставляется. Градация итоговых оценок для дисциплин с типами отчётности «экзамен» и «зачёт с оценкой» имеет следующий вид: 0–49 – «неудовлетворительно», 50–69 – «удовлетворительно», 70–89 – «хорошо», 90–100 – «отлично». Градация оценок для зачёта имеет следующий вид: 0–49 – «не зачтено», 50–100 – «зачтено». Все заносимые в систему баллы являются целыми, неотрицательными числами.

После перехода на страницу дисциплины на всех последующих страницах отображается навигационная цепочка, содержащая название выбранного ранее предмета, группы и страницы с её посещаемостью или оценками.

### **3.4.5 Студент**

После авторизации в системе студент попадает на страницу «еженедельное расписание». На этой странице отображается расписание студента на текущую неделю. Оно представлено в формате «разворота дневника»: слева – понедельник, вторник, среда; справа – четверг, пятница, суббота. Если размер экрана не позволяет отображать «разворот дневника», то дни недели располагаются друг под другом. В расписании указаны дата и время занятий, название дисциплины, номер аудитории (или «ДО», если занятие проходит в дистанционной форме), ФИО преподавателя, пункт для

отображения посещаемости. При нажатии на ФИО преподавателя, открывается всплывающее окно с полным ФИО и контактными данными этого преподавателя. В пункте для отображения посещаемости указан статус посещения студентом этого занятия. Эти данные вносятся преподавателем на странице «учёт посещаемости по дисциплине». Студент также может переключать недели назад и вперёд. Также на этой странице есть кнопка «код посещаемости», при нажатии которой открывается всплывающее окно, содержащее в себе поле для ввода кода и кнопка «отметить».

С этой страницы возможен переход на страницу «моя успеваемость», на которой отображена таблица с текущей успеваемостью студента. В таблице есть название дисциплины, ФИО преподавателя, оценки за три контрольные точки (аттестации), средний балл за три контрольные точки (аттестации), оценка за экзамен, итоговая оценка. Аналогично предыдущей странице, на этой также находится кнопка «код посещаемости».

### **3.5 Требования, не касающиеся функциональной части**

Разрабатываемое веб-приложение должно иметь архитектуру «клиент–сервер». Также необходимо реализовать взаимодействие front-end и back-end частей веб-приложения согласно архитектурному стилю REST.

Интерфейс веб-приложения необходимо оформить на русском языке. Иные языковые версии, помимо русскоязычной, не предусмотрены.

Все страницы веб-приложения должны быть выполнены в едином стиле. Вёрстка должна корректно отображаться в следующих браузерах:

- Google Chrome (версия desktop 95 или новее);
- Microsoft Edge (версия desktop 95 или новее);
- Opera (версия desktop 90 или новее).

Также корректное отображение веб-приложения на мобильных устройствах гарантируется в браузере Google Chrome (версия mobile 95 или новее).