Министерство науки и образования Российской федерации ФБГОУ ВО «Кубанский государственный технологический университет» (ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности	
Кафедра информационных систем и программирования	
Направление подготовки/специальность 09.03.04 Программная инжен	ерия
(код и наименование направления подготовки/сп	вециальности)
Профиль/специализация Проектирование и разработка программного обес (наименование профиля/специализации)	спечения
КУРСОВОЙ ПРОЕКТ	
по дисциплине Технологии разработки программного обеспечения (наименование дисциплины)	,
на тему: «Разработка образовательного веб-сервиса. Модуль клиент-серве	рного
ВЗАИМОДЕЙСТВИЯ» (тема курсовой работы)	
• • • • • • • • • • • • • • • • • • • •	
Выполнил(-а) студент(-ка) Сычев Федор Вячеславович курса 2 группы (фамилия, имя, отчество)	19-КБ-ПР1
Допущена к защите 15.12. 2020 (дата)	
Руководитель (нормоконтролер) работы	Попова О.Б.
Туководитель (нормоконтролер) работы (должность, подпись, дата)	
Защищена <u>15.12.2020</u> Оценка <u>От</u>	
Члены комиссии:	Кушнир Н.В.
Kul	Тотухов К.Е.
(должность, подпись, дата, расшифровка подписи)	

институт <u>компьютерных систем і</u>	и информационной оезопасности
Кафедра <u>информационных систе</u>	м и программирования
Направление подготовки/специал	изыность 09.03.04 Программная инженерия (код и наименование направления подготовки/специальности)
Профиль/специализация Проек	тирование и разработка программного обеспечения (наименование профиля/специализации)
	УТВЕРЖДАЮ
	Зав. кафедрой доц Янаева М.В.
	- Cullrugt
	« <u>В</u> 8» <u>О</u> 9 2020 г.
	ЗАДАНИЕ
	на курсовой проект
	Вячеславовичу курса 2 группы 19-КБ-ПР1
тема раооты: «Разраоотка образованию взаимодействия,»	овательного веб-сервиса. Модуль клиент-серверного
	ра института № от20 г.)
	a momiting a 31 or
План работы:	
 Изучение предметной области Проектирование баз данных. 	
	мирование запросов для взаимодействия системы с базой
данных.	inposonine ourpoos Ass. sometime and the second
OF THE POPULATION	
Объем работы:	65 c
б) иллюстративная часть	c. с. пист(-ов)
Рекомендуемая литература: 1. Роберт А. Максимчук. UML д	ING INOCTLY CMENTULIY
2 Йордон Элвард. Объектно-от	риентированный анализ и проектирование систем/ Эдвард
Йордон, Карл Аргила. – М.: ЛО	РИ, 2014. — 264 с.
Срок выполнения:	с « <u>01</u> » <u>09</u> по « <u>30</u> » <u>12</u> 2020 г.
Срок защиты:	« <u>21</u> » <u>12</u> 2020 г.
Дата выдачи задания:	« <u>ОВ</u> » <u>ОЯ</u> 2020 г.
Дата сдачи работы на кафедру:	<u>21</u> » <u>12</u> 2020 г.
Руководитель работы	доц_ (дояжноеть, подпись) Попова О.Б.
Задание принял студент	Сычев Ф.В.

Реферат

Пояснительная записка курсового проекта 65 с., 21 рис., 8 источников, 1 таблица, 2 приложения.

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ОБРАЗОВАТЕЛЬНЫЙ ВЕБ СЕРВИС, ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ, ДИАГРАММА КЛАССОВ, PHP, HTML, CSS, SASS, SQL, БАЗА ДАННЫХ MYSQL

Объектом исследования являются информационные потоки образовательных организаций.

Целью курсового проекта является проектирование и разработка программного продукта (системы) для управления образовательным процессом.

К полученным результатам относится разработанная система (сайт), которая предоставляет возможность управления образовательным процессом.

Содержание

Введение	6
1 Нормативные ссылки	8
2 Анализ предметной области	9
2.1 Цель приложения	9
2.2 Роли	9
2.3 Истории	9
2.4 Диаграмма вариантов использования	10
2.5 Аналоги	11
3 Техническое задание	13
3.1 Введение	13
3.2 Основания для разработки	13
3.3 Назначение разработки и область применения программы	13
3.4 Требование к программе или программному изделию	13
3.5 Требования к надежности	14
3.6 Время восстановления программы после отказа	14
3.7 Информационное обеспечение	15
3.8 Требования к базе данных	15
3.9 Требования к квалификации и численности персонала	15
3.10 Требования к составу и параметрам технических средств	16
3.11 Предварительный состав программной документации	16
3.12 Экономические преимущества разработки	16
3.13 Стадии разработки	17
3.14 Этапы разработки	17
4 Проектирование программного обеспечения	18
4.1 Функциональное проектирование	18
4.2 Проектирование интерфейса программы	19
4.2.1 Интерфейс авторизации/регистрации	19
4.2.2 Интерфейс основного окна приложения	20
4.3 Выбор языка программирования	21
4.4 Проектирование базы данных	
4.5 Проектирование классов	24

5 Реализация программного продукта	26
5.1 Описание методов программирования	26
5.2 Руководство пользователя	27
Заключение	35
Список используемых источников	36
Приложение А	37
Приложение Б	65

Введение

Целью курсового проекта является проектирование и разработка программного продукта, который будет реализовывать процесс создания, загрузки и прохождения уроков студентами, а также дальнейшее оценивание их результатов.

Задачи курсового проекта:

- изучить предметную область и провести анализ существующих решений;
 - составить техническое задание проекта;
 - спроектировать программный продукт;
 - реализовать программный продукт;
 - составить руководство пользователя.

Предметом исследования курсового проекта является разработка сайта на языках html и css, взаимодействующего с базой данных, созданной на языке SQL.

В первой главе проекта указаны нормативные ссылки, которые были использованы при написании курсового проекта.

Во второй главе перечислены основные цель разработки, форма ее реализации, перечень потребностей конечного пользователя приложения. В конце главы представлена диаграмма вариантов использования приложения. Разработку данной главы осуществляли оба участника Сычев Федор и Шишков Данил.

В третьей главе содержится техническое задание проекта, на основе которого строится работа учебного комплекса. Разработку данной главы осуществлял Сычев Федор.

В четвертой главе описан процесс проектирования программного продукта, выбор средств разработки. Разработку данной главы осуществляли оба участника Сычев Федор и Шишков Данил.

В пятой главе описан процесс разработки программного продукта и руководство пользования готовым программным продуктом. Разработку данной главы осуществлял Шишков Данил.

В заключении приводятся основные выводы из проведенного исследования, а также основные рекомендации по использованию сайта.

1 Нормативные ссылки

В данном курсовом проекте использованы следующие нормативные ссылки:

ГОСТ Р 1.5-2004 Стандартизация в Российской Федерации. Стандарты национальные Российской Федерации. Правила построения, изложения, оформления и обозначения.

ГОСТ Р 1.12-2004 Стандартизация в Российской Федерации. Термины и определения.

ГОСТ Р 7.0.5-2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления.

ГОСТ Р ИСО 9000-2008 Системы менеджмента качества. Основные положения и словарь.

ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств.

ГОСТ 2.001-93 ЕСКД. Общие положения.

ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных продуктов.

ГОСТ 19.105-78 ЕСПД. Обозначение программ и программных продуктов.

ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению.

ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.

2 Анализ предметной области

2.1 Цель приложения

Education Web-Servise – сайт для управления образовательным процессом.

Ожидается, что сайт будет использоваться в целях улучшения качества обучения на лекционно-практическом, а также лабораторном курсе студентов кафедры ИСП.

2.2 Роли

Роли показывают кем может быть человек в данном приложении и то, каким образом в дальнейшем он с ним будет взаимодействовать. В данном приложении существуют следующие роли:

- Те, кто использует сайт для прохождения уроков и просмотра лекций – «Студент»;
- Те, кто использует сайт для создания лекций и уроков для студентов «Преподаватель»;
- Те, кто поддерживают стабильную работу, оптимизируют и исправляют ошибки – «Техническая поддержка».
- Сайту необходим «Администратор», который будет консультировать «пользователя» по всем вопросам и проверять аккаунты новых пользователей.

2.3 Истории

Истории отображают действия, которые может выполнять человек соответствующей роли, они являются основными функциями приложения. В данном приложении представлены следующие истории:

- как студент я могу авторизовываться на сайте под моей учетной записью;
 - как студент я могу просматривать список дисциплин, уроков;

- как студент я могу проходить выбранные уроки;
- как студент я могу откладывать просмотр уроков или же просматривать их многократно;
- как студент я могу выполнять практические задания к урокам и видеть свои результаты с оценкой;
- как преподаватель я могу авторизовываться под своей учетной записью;
- как преподаватель я могу просматривать список студентов и групп, в которых я веду дисциплины;
- как преподаватель я могу просматривать список оценок, которые получили студенты после выполнения практических работ;
- как преподаватель я могу создавать уроки и практические задания;
- как преподаватель я могу при необходимости редактировать уже сделанные практические задания;
- как преподаватель я могу просматривать созданные мной уроки и практические задания.

2.4 Диаграмма вариантов использования

Вариант использования (use case) представляет собой последовательность действий, выполняемых системой в ответ на событие, инициализируемое некоторым внешним объектом. Вариант использования описывает типичное взаимодействие между пользователем и системой. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать. Диаграмма вариантов использования изображена на рисунке 2.4.1.

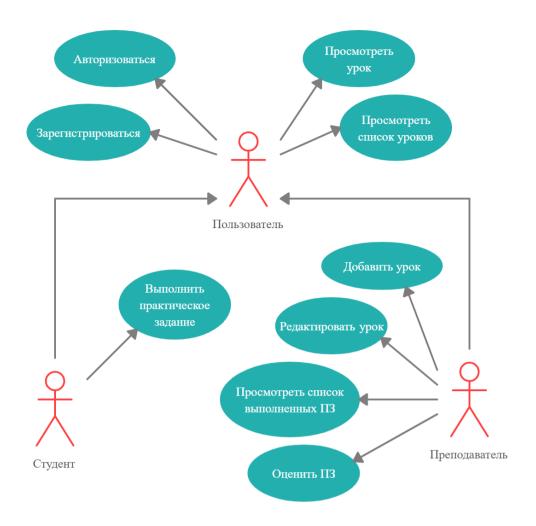


Рисунок 2.4.1 – Диаграмма вариантов использования

Диаграмма вариантов использования была построена при помощи ресурса [2].

2.5 Аналоги

Из аналогов данного приложения можно выделить образовательный портал geekbrains или Модульная объектно-ориентированная динамическая учебная среда КубГТУ moodle.kubstu, которые также представляют собой образовательные сервисы с возможностями добавления уроков, практических заданий и другими функциями. Рассмотрим их интерфейсы. На рисунке 2.5.1 представлен интерфейс сайта geekbrains.

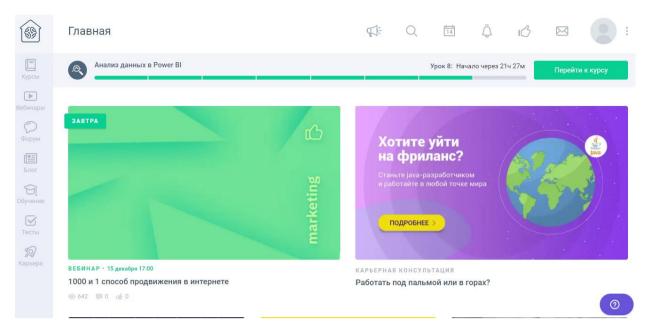


Рисунок 2.5.1 – Интерфейс сайта geekbrains

Теперь рассмотрим интерфейс сайта moodle.kubstu.ru. Его интерфейс показан на рисунке 2.5.2.

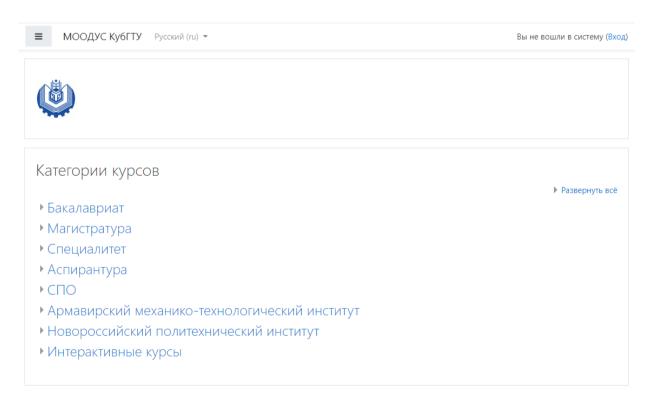


Рисунок 2.5.1 – Интерфейс сайта moodle.kubstu.ru

3 Техническое задание

3.1 Введение

Настоящее техническое задание распространяется на разработку образовательного веб сервиса.

Разрабатываемый сайт позволит пользователям использовать компьютер, планшет или мобильный телефон для прохождения уроков, а также выполнения практических заданий.

3.2 Основания для разработки

Указание о закреплении тем курсовых проектов №24-КТ от 23.09.2020.

3.3 Назначение разработки и область применения программы

Функциональным назначением программного продукта является предоставление пользователю возможности дистанционного просмотра лекций, уроков, а также практических заданий по различным дисциплинам.

Программа должна эксплуатироваться студентами и преподавателями в практическом и лабораторном курсе дисциплин кафедры ИСП. Программа позволит повысить уровень и качество учебного процесса студентов кафедры ИСП, упростить процесс создания уроков и практических заданий преподавателями и процесс прохождения созданных уроков студентами.

3.4 Требование к программе или программному изделию

Программа должна предоставлять пользователю следующий функционал:

- авторизация в качестве преподавателя или студента;
- создание преподавателем уроков и практических заданий;

- редактирование преподавателем уроков и практических заданий;
- при создании практического задания преподаватель может выбирать тип задания (с коротким ответом, с развернутым ответом, с кодом программы);
- просмотр всех групп и студентов, чьи дисциплины ведет данный преподаватель;
- просмотр результатов выполненных практических работ студентов, их оценок, а также прогресс по просматриваемым урокам;
 - комментирование результатов студентов;
 - выполнение практических заданий студентами;
 - просмотр уроков студентами;
 - просмотр информации о себе и о своей группе;
 - просмотр на главной странице расписания на следующие 2 дня.

3.5 Требования к надежности

- ПО должно иметь защиту от некорректных действий пользователей и ошибочных исходных данных;
- ПО не должно во время работы модифицировать свой код или код других программ;
- количество отказов ПО из-за не выявленных ошибок не должно превышать 1 отказа на 1000 сеансов работы с программой.

3.6 Время восстановления программы после отказа

- максимальное время устранения неисправностей в работе оборудования сервера системы 24 часа;
- максимальное время устранения неисправностей в работе по сервера системы – 12 часов;

- максимальное время устранения неисправностей в работе sql сервера бд системы 12 часов;
- время устранения сбоев в работе системного и прикладного по рабочих станций – 12 часов.

3.7 Информационное обеспечение

- средства ввода данных в систему должны обеспечивать контроль правильности данных по типу;
- при модификации и удалении данных средства ведения ПО должны запрашивать подтверждение правильности выданных команд и временно сохранять предыдущую версию данных;
- при выполнении запросов, время которых превышает 10 секунд,
 на монитор пользователя должен выдаваться транспарант с извещением о вынужденном ожидании.

3.8 Требования к базе данных

- должна быть 3 или 4 нормальной формы;
- ограничение на размер загружаемого файла 50 мб;
- быстрый поиск по ключу;
- отсутствие избыточных данных;
- защита данных от несанкционированного доступа;
- простота добавления студента или преподавателя в базу данных.

3.9 Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы должно составлять не менее 2 штатных единиц – системный

администратор и конечный пользователь программы – оператор. В перечень задач, выполняемых системным администратором, входит:

- задача поддержания работоспособности технических средств;
- задачи установки (инсталляции) и поддержания работоспособности системных программных средств операционной системы;
 - задача создания резервных копий базы данных.

3.10 Требования к составу и параметрам технических средств

В состав технических средств должны входить IBM-совместимый персональный компьютер, выполняющий роль сервера, включающий в себя:

- 2-разрядный или 64-разрядный процессор с тактовой частотой 1
 ГГц или выше;
- 1 ГБ ОЗУ для 32-разрядного процессора или 2 ГБ ОЗУ для 64разрядного процессора;
- 16 ГБ для 32-рязрядной системы или 20 ГБ для 64-разрядной системы свободного места на жестком диске;
 - операционную систему Windows 7 или новее;
 - apache http-сервер.

3.11 Предварительный состав программной документации

Состав программной документации должен включать в себя:

- техническое задание;
- программный продукт и методику испытаний;
- руководство оператора.

3.12 Экономические преимущества разработки

Ориентировочная экономическая эффективность не рассчитывается в виду уникальности предъявляемых требований к разработке.

3.13 Стадии разработки

Разработка должна быть проведена в 3 стадии:

- разработка технического задания;
- рабочее проектирование;
- внедрение.

3.14 Этапы разработки

На стадии разработки ДЗ должен быть выполнен этап разработки, согласования и утверждение технического задания. На стадии рабочего проектирования должны быть выполнены следующие этапы работ:

- разработка программы;
- разработка программной документации;
- испытания программы.

4 Проектирование программного обеспечения

4.1 Функциональное проектирование

На основе технического заданий построим функциональные диаграммы, которые изображены на рисунках 4.1.1 и 4.1.2.

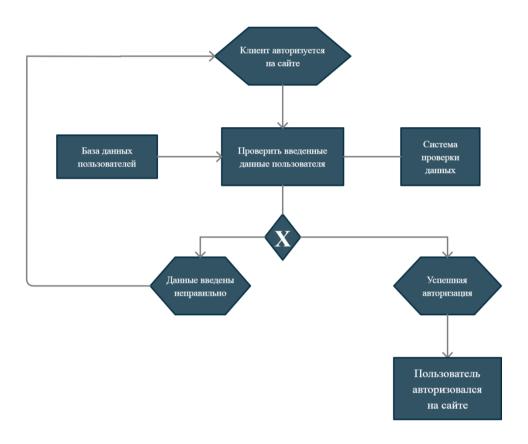


Рисунок 4.1.1 – функциональная диаграмма авторизации на сайте

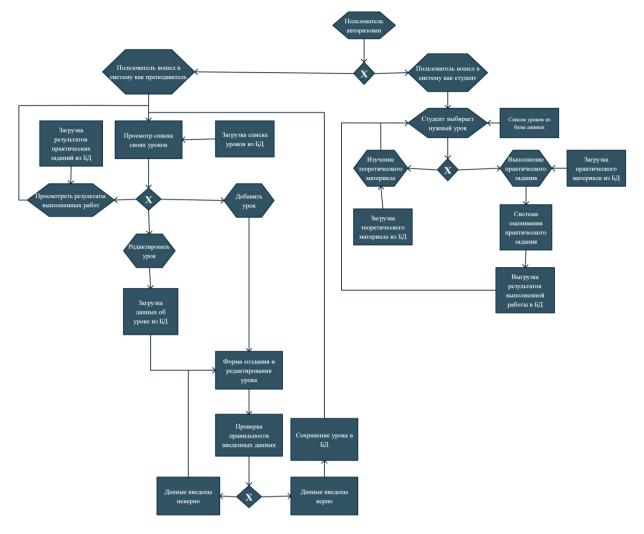


Рисунок 4.1.2 – Функциональная диаграмма сайта

На функциональных диаграммах отображена вся последовательность действий сайта, а также его взаимодейтствие с базой данных во всевозможных вариантах событий.

4.2 Проектирование интерфейса программы

4.2.1 Интерфейс авторизации/регистрации

Каждый новый пользователь программы должен перед ее использованием зарегистрировать себя, создать себе аккаунт, на который будут приходить уведомления, сообщения и т.д. И так-же те, кто уже пользовался услугами данной системы и соответственно использовал ее, должны авторизоваться, чтобы получить доступ к своим данным. Макет интерфейса формы продемонстрирован на рисунке 4.2.1.1.

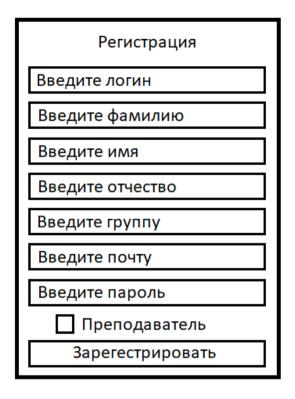


Рисунок 4.2.1.1 – Макет интерфейса формы регистрации

4.2.2 Интерфейс основного окна приложения

После регистрации и авторизации пользователя в системе, открывается основная страница сайта. Макет интерфейса основной страницы сайта изображен на рисунке 4.2.2.1.

Информация о пользователе Расписание Практические задания

Рисунок 4.1.2.1 – Макет интерфейса основного окна приложения

4.3 Выбор языка программирования

Для написания данного проекта будет использоваться язык программирования PHP, язык гипертекстовой разметки HTML, база данных MySQL.

4.4 Проектирование базы данных

Далее необходимо спроектировать базу данных. Для успешного решения возложенных на систему задач рассматриваемая модель должна включать следующие элементы:

- студенты;
- преподаватели;
- профили подготовки;
- факультеты;
- группы;
- уроки;

- практические задания;
- дисциплины.
- Взаимодействие и взаимосвязь выделенных объектов базируются на следующих выявленных в результате обследования ПО концепциях:
- студентов объединяет профиль обучения, к которому они относятся;
 - профили обучения соотносятся со специальностями студентов;
 - каждый профиль подготовки объединяет преподавателей;
- каждому преподавателю присваивается список дисциплин,
 которые он ведет;
- по данной специальности студенты приказом зачисляются в определенную группу. каждая группа включает много студентов;
- каждому студенту присваиваются все файлы, которые он загрузил для хранения на сервере;
- преподаватель ведет определенные дисциплины, создает уроки и практические задания, которые связываются с этой дисциплиной.

Для каждого объекта определим необходимые атрибуты. Изучение состава атрибутов каждого объекта позволило выделить те из них, которые можно использовать в качестве первичных ключей. Результаты анализа данной предметной области приведены в таблице 1.

Таблица 1 – Проектирование базы данных.

ОБЪЕКТ	АТРИБУТ	ПЕРВИЧНЫЙ КЛЮЧ
ПРОФИЛЬ ПОДГОТОВКИ	Id_профиля_подготовки Название	Id_профиля_подготовки
ФАЙЛ СТУДЕНТА	Id_файла Имя_файла Путь_к_файлу Id_преподавателя Id_студента Дисциплина	Id_файла

	Дата_создания Дата_обновления	
ПРЕПОДАВ АТЕЛЬ	Дата_ооновления Id_профиля Фамилия Имя Отчество	Id_профиля
СПЕЦИАЛЬ НОСТЬ	Id_специальности Наименование_ специальности	Id_специальности
ГРУППА	Id_группы Название_группы	Id_группы
СТУДЕНТ	Id_профиля Фамилия Имя Отчество Id_направления	Id_профиля
УРОК	Id_урока Id_дисциплины Id_преподавателя Содержание урока	Id_урока
ПРАКТИЧЕ СКАЯ РАБОТА	Id_практической Id_урока Задание Материал Ответы	Id_практической

После проектирования базы данных создадим ER-диаграмму разрабатываемой базы данных. ER-диаграмма приведена на рисунке 4.3.1. добавить таблицы Discussions Также базу ОНЖОМ В данных И Message discussions для того, чтобы заранее предусмотреть возможность создания обсуждений студентами в случае, если у кого-то останутся вопросы по теме урока.

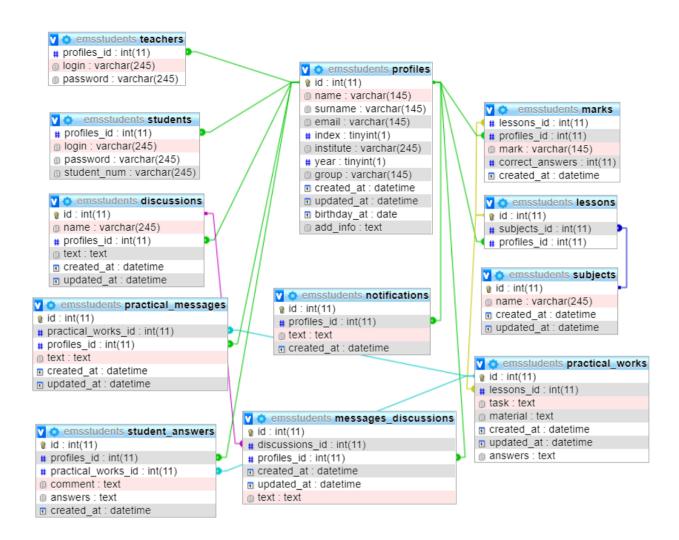


Рисунок 4.3.1 – ER-диаграмма базы данных

4.5 Проектирование классов

На сайте основным объектом будет пользователь, он может быть двух типов — студент или преподаватель, для этого нужно реализовать один базовый и два класса наследника. Далее на сайте будут присутствовать дисциплины, в них будут присутствовать лекции, или уроки, в которых должны быть практические задания, все эти классы так-же нужно реализовать. На рисунке 4.3.1 представлена диаграмма классов проекта.

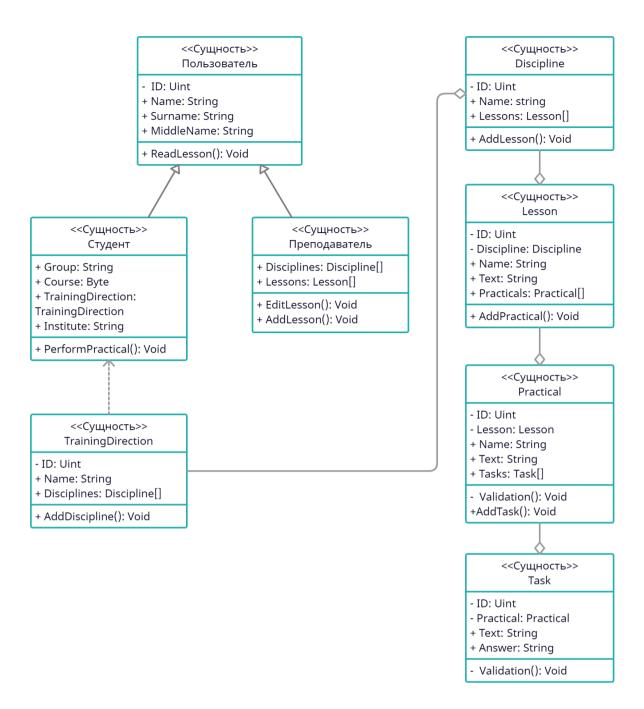


Рисунок 4.5.1 – Диаграмма классов

5 Реализация программного продукта

5.1 Описание методов программирования

Первое, что необходимо сделать сложной системе (сайту) — подключиться к своей базе данных. Подключение к БД реализовано в отдельном файле connect.php, для удобства и быстрого редактирования кода. Программный код подключения к БД:

```
$db = mysqli_connect('localhost', 'mysql', 'mysql', 'ewb');
if ($db == false)
    echo("<script>alert('He удалось подключиться к БД');</script>"
);
```

Информация по работе с методом mysqli_connect найдена на электронном ресурсе [5].

Для сохранения и получения данных из базы данных необходим метод отправления запросов в базу данных, это метод mysqli_query. Пример запроса в базу данных:

```
$result = mysqli_query($db, "SELECT `name`, `surname`, `middle_n
ame`, `institute`, `group`, `index` FROM `profiles` WHERE `id` = '$pro
files_id'");
```

Хранение данных об авторизованном пользователе производится в куки-файлах. Пример сохранения куки-данных:

```
setcookie('profiles_id', $profiles_id, time() + 3600 * 24 * 7, "
/");
```

Пример получения куки-данных:

```
$profiles id = $ COOKIE['profiles id'];
```

Информация по работе с куки-данными найдена на электронном ресурсе [7].

При работе с заполняемыми формами и полями необходимо передавать данные из этой формы в метод action, для этого существует два способа получения данных из полей GET и POST. Пример формы с методом передачи данных POST:

Получение данных из глобальной переменной POST:

GET данные можно записывать прямо в URL адрес страницы, пример записи данных в глобальную переменную GET:

```
<a href="lessons.php?subjects id=<?=$value['id']?>">
```

Получение данных из глобальной переменной GET:

```
$subjects_id = $_GET['subjects_id'];
```

Информация по работе с GET и POST данными найдена на электронном ресурсе [8].

Весь код программного продукта представлен в Приложении А.

5.2 Руководство пользователя

Первым делом пользователь должен произвести авторизацию, в случае если есть аккаунт, или регистрацию, когда аккаунта у пользователя еще нет. Для этого необходимо заполнить все поля регистрационной формы и нажать на кнопку «Зарегистрироваться» (пример формы изображен на рисунке 5.2.1). Если пользователь введет логин, который уже используется другим пользователем в данной системе, ему выведется сообщением об этом (см. пример на рисунке 5.2.2), для решения этой ошибки необходимо ввести другой логин.

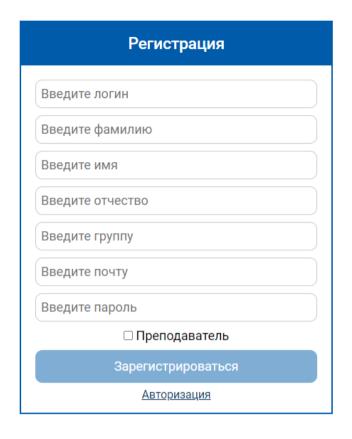


Рисунок 5.2.1 – Пример формы регистрации

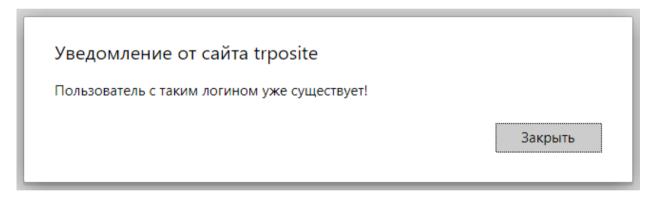


Рисунок 5.2.2 – Ошибка регистрации пользователя

В случае авторизации необходимо ввести логин и пароль, в случае, когда логин и/или пароль неверный, выведется сообщение об ошибке авторизации, как и при регистрации.

После успешной авторизации/регистрации пользователь попадает на главную страницу, где написаны все его данные (см. пример на рисунке 5.2.3).

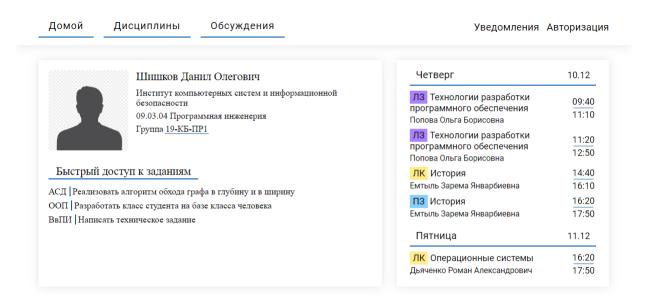


Рисунок 5.2.3 – Пример главной страницы

Сверху, в шапке сайта, можно увидеть вкладки «Домой» – это главная страница сайта (пример на рисунке 5.2.3), «Дисциплины» – вкладка со всеми представленными дисциплинами в системе, на ней можно выбрать одну из дисциплин для дальнейшего прохождения лекционного и практического занятия. Пример страницы с дисциплинами изображен на рисунке 5.2.4.

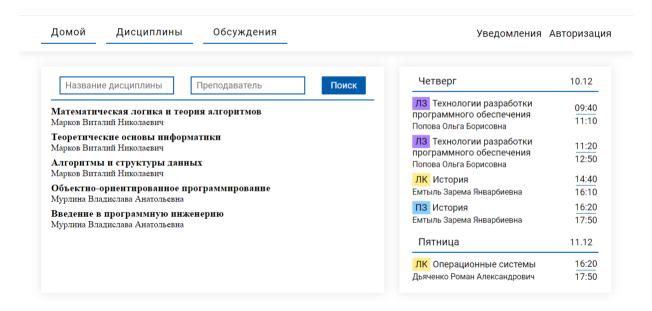


Рисунок 5.2.4 – Пример страницы с дисциплинами

После выбора одной из дисциплины пользователь попадает на страницу со списком лекций данной дисциплины. На рисунке 5.2.5

изображена страница с лекциями для дисциплины «Объектноориентированное программирование».

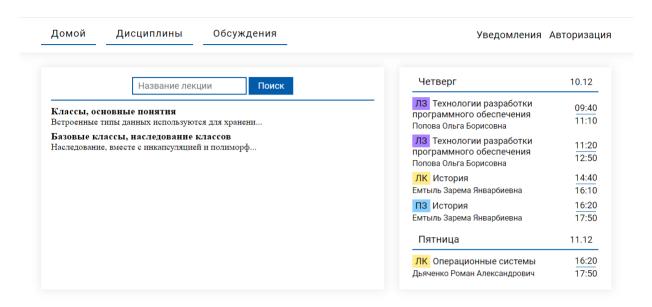


Рисунок 5.2.5 – Пример страницы с лекциями

После выбора лекции, пользователь попадает на страницу с материалом лекции. С которым он ознакомляется и может приступить к выполнение практических заданий, которые представлены в нижней части страницы (см. пример на рисунке 5.2.6). Пример страницы практического задания изображен на рисунке 5.2.7.

<u> Дисциплины</u> Обсуждения ————————————————————————————————————		Авторизац
Классы, основные понятия	Четверг	10.12
Встроенные типы данных используются для хранения одного значения в ыбъявленной переменной. Например, int х хранит целое значение в переменной с именем к. В объектно-ориентированном программировании, класс является типом (анных, определяющим набор переменных и методов для объявленного объекта. Запример, если необходимо создать программу, которая управляет счетами в банке, бо класс ВапкАссоци может быть использован для объявления объекта, который будет иметь все свойства и методы, необходимые для управления индивидуальным счетом банка, такие как переменную balance и методы Deposit и Withdrawal. Класс похож на шаблон. Он определяет данные и поведение типа. Определение класса начинается с ключевого слова class, за которым следует имя класса. Тело класса посрежит данные и действия, заключенные в фигурные скобки. Так же, как и истроенные типы могут быть использованы для объявления множества истроенных, классы могут быть использованы для объявления множества истроенных. Классы могут быть использованы для объявления множества истроенных класса (разрабатываем) класс, который является шаблоном для объектов. Мы определяем (разрабатываем) класс, который является шаблоном для которым валасса) и производится с использованием ключевого слова пем. Саждый объект имеет свои собственные характеристики - поля. Значения этих полей описывают текущее состояние объекта. Например, Person (объект класса сетезоп) может быть в возрасте 30 лет, мужского поля, и называться Апtопію. Инкапсуляция Частью значения слова инкапсуляция значит больше, чем просто комбинирование порций функционала внутри классов; она также значит впраничение доступа к этому классу от внешнего воздействия. Инкапсуляция упраничение доступа к этому классу от внешнего воздействия. Инкапсуляция пределяют область видимости членов класса. С# поддерживает следующия пределяют область видимости членов класса. С# поддерживает следующия пременны класса доступными голько изнутри класса, и прячет их от доступа наменны класса доступными только изнутри класса, и прячет их от доступ	лз Технологии разработки программного обеспечения Попова Ольга Борисовна лз Технологии разработки программного обеспечения Попова Ольга Борисовна лк История Емтыль Зарема Январбиевна пз История Емтыль Зарема Январбиевна Пятница лк Операционные системы Дьяченко Роман Александрович	09:40 11:10 11:20 12:50 14:40 16:10 16:20 17:50 11.12

Рисунок 5.2.6 – Пример страницы лекционного материала

Модификаторы доступа

Домой <u>Дисциплины</u> Обсуждения ————————————————————————————————————	Уведомления А	≀вторизаци
Модификаторы доступа	Четверг	10.12
Какие три основных модификаторов доступа существует в С#? Перечислите через апятую.	ЛЗ Технологии разработки программного обеспечения Попова Ольга Борисовна	09:40 11:10
Введите ответ Отправить	ЛЗ Технологии разработки программного обеспечения Попова Ольга Борисовна	11:20 12:50
	ЛК История	14:40
	Емтыль Зарема Январбиевна	16:10
	ПЗ История	16:20
	Емтыль Зарема Январбиевна	17:50
	Пятница	11.12
	ЛК Операционные системы	16:20
	Дьяченко Роман Александрович	17:50

Рисунок 5.2.7 – Пример страницы практического задания

В случае, когда пользователь является преподавателем, интерфейс сайте немного изменяется. На страницах дисциплин, лекций и практических занятий появляются формы для добавления новых дисциплин, лекций и практических заданий. Примеры страниц с дополнительным интерфейсом изображены на рисунках 5.2.8-10.

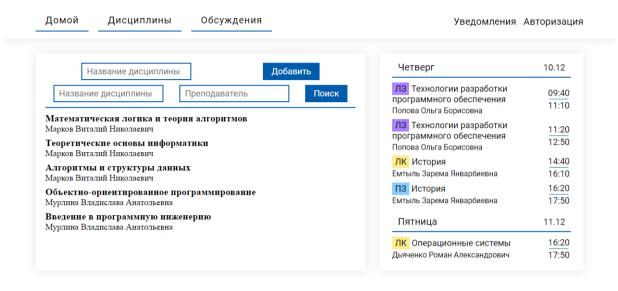


Рисунок 5.2.8

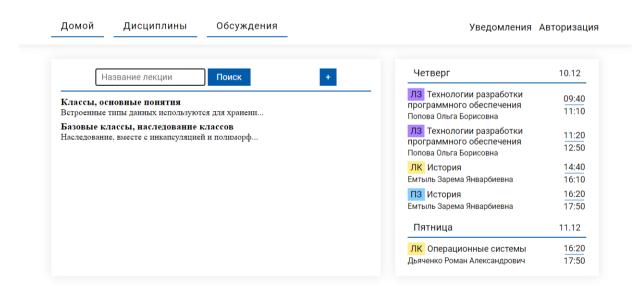


Рисунок 5.2.9

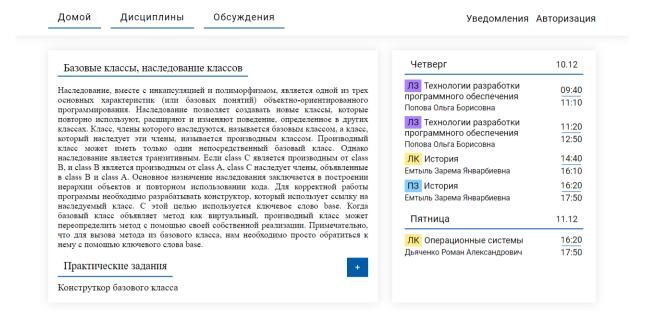


Рисунок 5.2.10

Для добавления новой лекции или практического задания необходимо нажать на кнопку с плюсиком. После чего вы попадете на страницу создания новой лекции или практического задания. Пример добавление новой лекции изображен на рисунке 5.2.11, а добавление нового практического задания на рисунке 5.2.12.

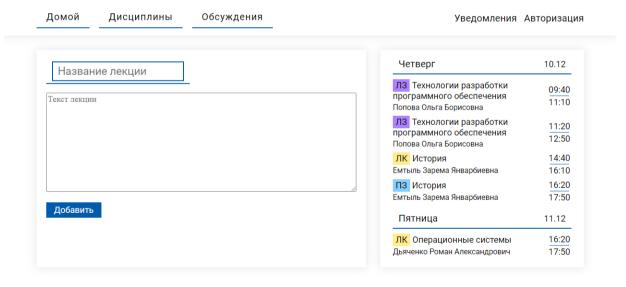


Рисунок 5.2.11 – Добавление новой лекции

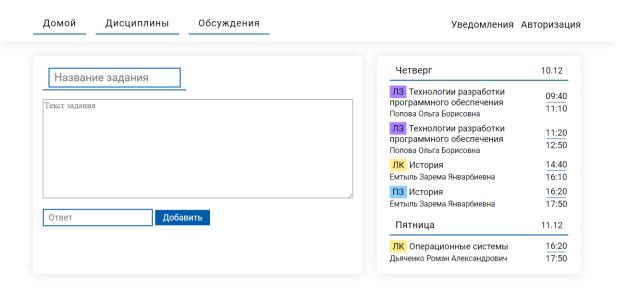


Рисунок 5.2.12 – Добавление нового практического задания

Заключение

В результате курсового проекта по дисциплине «Технологии разработки программного обеспечения» разработан сайт, позволяющий реализовывать процесс создания, загрузки и прохождения уроков студентами.

Цель курсового проекта, которая заключается в проектировании и разработке учебного комплекса, выполнена.

В ходе работы были выполнены следующие задачи:

- произведено изучение предметной области;
- произведен анализ существующих решений и аналогов;
- техническое задание проекта составлено;
- программный продукт спроектирован;
- учебный комплекс реализован;
- составлено руководство пользователя и пояснительная записка проекта.

В ходе разработки использовались такие программные средства, как язык программирования PHP, язык гипертекстовой разметки HTML, язык каскадных таблиц стилей CSS и препроцессор SASS к нему, а также использование базы данных MySQL, созданной в среде разработки MySQL Workbench

Список используемых источников

- 1. Техническое задание. Требование к содержанию и оформлению. ГОСТ 19.201-78 [Текстовый документ]
 - 2. Creately [Электронный ресурс] https://creately.com/
- 3. Образовательный портал GeekBrains [Электронный ресурс] https://geekbrains.ru
- 4. Модульная объектно-ориентированная динамическая учебная среда КубГТУ moodle.kubstu [Электронный ресурс] https://moodle.kubstu.ru
- 5. Mysqli_connect Manual [Электронный ресурс]https://www.php.net/manual/ru/function.mysqli-connect.php
- 6. Mysqli_query Manual [Электронный ресурс] https://www.php.net/manual/ru/mysqli.query.php
- 7. Работа с cookie [Электронный ресурс] https://metanit.com/web/php/4.2.php
- 8. PHP Методы GET и POST [Электронный ресурс] https://webformyself.com/php-metody-get-i-post/

Приложение А

Листинг

Файл authorization.php

```
<?php
  $message = $ COOKIE['message'];
  setcookie('message', "", time() - 60, "/authorization.php");
?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <?php include 'includes/head.php'; ?>
  <link rel="stylesheet" href="css/login.css">
</head>
<body>
<?php if ($message) echo("<script>alert('".$message."');</script</pre>
>"); ?>
  <div class="connect-wrap">
    <div class="connect">
      <h2 class="connect title">Авторизация</h2>
      <form class="connect form" action="includes/login.php" me</pre>
thod="post" name="user">
        <input type="text" name="login" placeholder="Введите лог
ин..." required autocomplete="on" autofocus>
        <input type="password" name="password" placeholder="Введ
ите пароль..." required>
        <input type="submit" value="Войти">
        <a class="connect else" href="registration.php">Регистр
ация</а>
      </form>
    </div>
  </div>
</body>
</html>
Файл check.php
<?php
  include 'connect.php';
  // Получаем данные с форм
  $isTeacher = ($ POST['isTeacher'] == "on") ? 1 : 0;
  $login = filter var(trim($ POST['login']), FILTER SANITIZE STR
ING);
  $name = filter var(trim($ POST['name']), FILTER SANITIZE STRIN
G);
```

```
$surname = filter var(trim($ POST['surname']), FILTER SANITIZE
  $middle name = filter var(trim($ POST['middle name']), FILTER
SANITIZE STRING);
  $group = $isTeacher ? null : filter var(trim($ POST['group']),
FILTER SANITIZE STRING);
  $email = filter var(trim($ POST['email']), FILTER SANITIZE STR
  $password = filter var(trim($ POST['password']), FILTER SANITI
ZE STRING);
 // Проверяем существует ли уже пользователь
  $result teachers = mysqli query($db, "SELECT * FROM teachers W
HERE `login` = '$login'");
  $result students = mysqli query($db, "SELECT * FROM students W
HERE `login` = '$login'");
  if (mysqli num rows($result students) > 0 || mysqli num rows($
result teachers) != 0) {
    setcookie('message', "Пользователь с таким логином уже сущес
TByeT!", time() + 60, "/registration.php");
   header('Location: /registration.php');
   exit;
  }
 // Сохраняем основные данные пользователя
 mysqli query($db, "INSERT INTO `profiles` (`name`, `surname`,
`middle_name`, `email`, `index`, `institute`, `year`, `group`) V
ALUES ('$name', '$surname', '$middle name', '$email', $isTeacher,
'MKCMB', 2, '$group')");
  $profiles id = mysqli insert id($db);
  // Сохраняем данные для входа студента/преподавателя
  $hash pass = md5($pass."educationwebservice");
  if ($isTeacher)
   mysqli query($db, "INSERT INTO `teachers` VALUES ('$profiles
id', '$login', '$hash pass')");
 else
   mysqli query($db, "INSERT INTO `students` VALUES ('$profiles
id', '$login', '$hash pass')");
 mysqli close($db);
 setcookie('profiles id', $profiles id, time() + 3600 * 24 * 7,
"/");
 header('Location: /');
?>
Файл registration.php
<?php
```

```
$message = $ COOKIE['message'];
  setcookie('message', "", time() - 60, "/registration.php");
?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <?php include 'includes/head.php'; ?>
  <link rel="stylesheet" href="css/login.css">
</head>
<body>
  <?php if ($message) echo("<script>alert('".$message."');</scri</pre>
pt>"); ?>
  <div class="connect-wrap">
    <div class="connect">
      <h2 class="connect title">Регистрация</h2>
      <form class="connect form" action="includes/check.php" me</pre>
thod="post" name="user">
        <input type="text" name="login" placeholder="Введите лог
ин" required autocomplete="on" autofocus>
        <input type="text" name="surname" placeholder="Введите ф
амилию" required>
        <input type="text" name="name" placeholder="Введите имя"</pre>
required>
        <input type="text" name="middle name" placeholder="Введи
те отчество" required>
        <input type="text" name="group" placeholder="Введите гру
ппу" required>
        <input type="text" name="email" placeholder="Введите поч
Ty" required>
        <input type="password" name="password" placeholder="Введ
ите пароль" required>
        <label><input type="checkbox" name="isTeacher"> Преподав
aтель</label>
        <input type="submit" value="Зарегистрироваться">
        <a class="connect else" href="authorization.php">Автори
зация</а>
      </form>
    </div>
  </div>
</body>
</html>
Файл login.php
<?php
  include 'connect.php';
  // Получаем данные с форм
```

```
$login = filter var(trim($ POST['login']), FILTER SANITIZE STR
  $password = filter var(trim($ POST['password']), FILTER SANITI
ZE STRING);
  $hash pass = md5($pass."educationwebservice");
  $result teachers = mysqli query($db, "SELECT * FROM teachers W
HERE `login` = '$login' AND `password` = '$hash pass'");
  $result students = mysqli query($db, "SELECT * FROM students W
HERE `login` = '$login' AND `password` = '$hash pass'");
  if (mysgli num rows ($result teachers) == 0 && mysgli num rows (
$result students) == 0) {
    setcookie('message', "Пользователь не найден!", time() + 60,
 "/authorization.php");
    header('Location: /authorization.php');
    exit();
  }
 mysqli close($db);
  if (mysqli num rows($result teachers) > 0)
    $profiles id = mysqli fetch assoc($result teachers)['profile
s id'];
  else
    $profiles id = mysqli fetch assoc($result students)['profile
  setcookie('profiles id', $profiles id, time() + 3600 * 24 * 7,
 header('Location: /');
?>
Файл isLogin.php
<?php
  if ($ COOKIE['profiles id'] == '')
    header('Location: authorization.php');
  $profiles id = $ COOKIE['profiles id'];
  include 'includes/connect.php';
  $result = mysqli query($db, "SELECT `name`, `surname`, `middle
_name`, `institute`, `group`, `index` FROM `profiles` WHERE `id`
= '$profiles id'");
  if (mysqli num rows($result) == 0)
   header('Location: authorization.php');
  $user = mysqli fetch array($result);
?>
Файл connect.php
<?php
$db = mysqli connect('localhost', 'mysql', 'mysql', 'ewb');
```

```
if ($db == false) {
  echo("<script>alert('He удалось подключиться к БД');</script>"
);
}
?>
```

Файл index.php

```
<?php
  include 'includes/isLogin.php';
?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <?php include 'includes/head.php'; ?>
  <link rel="stylesheet" href="css/schedule.css">
  <link rel="stylesheet" href="css/home.css">
</head>
<body>
  <?php include 'includes/header.php'; ?>
  <div class="container flex-row">
    <main class="main" id="main">
      <section class="person">
        <div class="person photo">
          <img src="/img/person-photo.jpg" alt="Фото">
        <div class="person-content">
          <h2 class="person-
content name"><?php echo($user['surname']." ".$user['name']." "</pre>
.$user['middle name']); ?></h2>
          <div>Институт компьютерных систем и информационной без
oпасности</div>
          <?php if ($user['index'] == 0):?>
          <div>09.03.04 Программная инженерия</div>
          <div class="person-
content group">Γρуππα <a href="#" class="underline"><?php echo(
$user['group']); ?></a></div>
          <?php endif; ?>
        </div>
      </section>
      <section class="tasks">
        <h2 class="tasks header"><span>Быстрый доступ к задания
M</span></h2>
```

```
<a href="">
             <span class="abbreviation">ACД</span>
            Реализовать алгоритм обхода графа в глубину и в ши
рину
           </a>
         <a href="">
            <span class="abbreviation">00П</span>
            Разработать класс студента на базе класса человека
           </a>
         <a href="">
            <span class="abbreviation">ВвПИ</span>
            Написать техническое задание
           </a>
         </section>
   </main>
   <?php include 'includes/sidebar.php'; ?>
 </div>
 <?php include 'includes/footer.php'; ?>
</body>
</html>
Файл discipline.php
<?php
 include 'includes/isLogin.php';
 // Проверяем на преподавателя
 $isTeacher = $user['index'] == 1;
 // Загружаем список дисциплин
 $result = mysqli query($db, "SELECT id, name, teacher id, (SEL
ECT name AS teacher name FROM profiles WHERE id = teacher id) AS
teacher name, (SELECT surname FROM profiles WHERE id = teacher
id) AS teacher surname, (SELECT middle name FROM profiles WHERE
id = teacher id) AS teacher middle name FROM subjects;");
 $subjects = mysqli fetch all($result, MYSQLI ASSOC);
 mysqli close($db);
?>
<!DOCTYPE html>
<html lang="ru">
```

```
<head>
  <?php include 'includes/head.php'; ?>
  <link rel="stylesheet" href="css/schedule.css">
  <link rel="stylesheet" href="css/discipline.css">
</head>
<body>
  <?php include 'includes/header.php'; ?>
  <div class="container flex-row">
    <main class="main" id="main">
      <section class="search">
        <?php if ($isTeacher): ?>
        <form class="discipline add" action="add.php?add=discipl</pre>
ine" method="post" name="discipline add">
         <input type="text" name="title" placeholder="Название</pre>
дисциплины" required autocomplete="on" autofocus>
         <input type="submit" value="Добавить">
        </form>
        <?php endif;?>
        <form class="search-
field" action="discipline.php" method="post" name="discipline">
         <input type="text" name="name" placeholder="Название д
исциплины" required autocomplete="on" autofocus>
         <input type="text" name="teacher" placeholder="Препода
ватель" required>
         <input type="submit" value="Поиск">
        </form>
        <?php foreach ($subjects as $value) { ?>
         <a href="lessons.php?subjects id=<?=$value['id']?>">
             <div class="discipline">
               <h3 class="discipline-
title bold"><?=$value['name']?></h3>
               <div class="discipline-teacher">
                 <?=$value['teacher surname'] . " " . $value['t</pre>
eacher name'] . " " . $value['teacher middle name']?>
               </div>
             </div>
           </a>
         <?php } ?>
        </111>
      </section>
    </main>
    <?php include 'includes/sidebar.php'; ?>
  </div>
```

```
<?php include 'includes/footer.php'; ?>
</body>
</html>
```

Файл lessons.php

```
<?php
  include 'includes/isLogin.php';
  include 'includes/connect.php';
  $subjects id = $ GET['subjects id'];
  $profiles id = $ COOKIE['profiles id'];
  // Проверяем на преподавателя
  $result = mysqli query($db, "SELECT * FROM `subjects` WHERE `i
d` = '$subjects id' AND `teacher id` = '$profiles id'");
  $isTeacherThis = mysqli num rows($result) > 0;
  $result = mysqli query($db, "SELECT `id`, `title`, `text` FROM
 `lessons` WHERE `subjects id` = '$subjects id';");
  $lessons = mysqli fetch all($result, MYSQLI ASSOC);
 mysqli close($db);
?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <?php include 'includes/head.php'; ?>
  <link rel="stylesheet" href="css/schedule.css">
  <link rel="stylesheet" href="css/discipline.css">
</head>
<body>
  <?php include 'includes/header.php'; ?>
  <div class="container flex-row">
    <main class="main" id="main">
      <section class="search">
        <div class="search-field">
          <form action="lessons.php" method="post" name="lessons</pre>
">
            <input type="text" name="name" placeholder="Название</pre>
лекции" required autocomplete="on" autofocus>
            <input type="submit" value="Поиск">
          </form>
          <?php if ($isTeacherThis): ?>
            <a class="add_button" href="lesson add.php?subjects</pre>
id=<?=$subjects id?>">+</a>
          <?php endif; ?>
```

```
</div>
       <?php if ($lessons) foreach ($lessons as $lesson) { ?>
        <a href="lesson.php?lesson id=<?=$lesson['id']?>">
            <div class="lesson">
              <h3 class="lesson-title bold">
                <?=$lesson['title']?>
              </h3>
              <div class="lesson-subtitle">
                <?php echo (mb strimwidth($lesson['text'], 0,</pre>
50, '...')); ?>
              </div>
            </div>
          </a>
        <?php } ?>
       </section>
   </main>
   <?php include 'includes/sidebar.php'; ?>
 </div>
 <?php include 'includes/footer.php'; ?>
</body>
</html>
```

Файл lesson.php

```
$practicals = mysqli fetch all($result, MYSQLI ASSOC);
 mysqli close($db);
?>
<!DOCTYPE html>
<html lang="ru">
<head>
 <?php include 'includes/head.php'; ?>
 <link rel="stylesheet" href="css/schedule.css">
 <link rel="stylesheet" href="css/lesson.css">
</head>
<body>
 <?php include 'includes/header.php'; ?>
 <div class="container flex-row">
   <main class="main" id="main">
     <section class="lesson">
       <div class="lesson title">
         <h2>
         <?=$lesson['title']?>
         </h2>
       </div>
       <?=$lesson['text']?>
       </section>
     <section class="practicals">
       <div class="practicals title">
         <h2>Практические задания</h2>
         <?php if ($isTeacherThis): ?>
           <a class="add button" href="practical add.php?lesson</pre>
id=<?=$lesson id?>">+</a>
         <?php endif; ?>
       </div>
       <?php if ($practicals) foreach ($practicals as $practica</pre>
1) { ?>
         <a href="practical.php?practical id=<?=$practical['i</pre>
d'1?>">
             <div class="practical">
               <h3 class="practical-title">
                 <?=$practical['task']?>
               </h3>
             </div>
           </a>
```

Файл practical.php

```
<?php
 include 'includes/isLogin.php';
 include 'includes/connect.php';
 $practical id = $ GET['practical id'];
 $result = mysqli query($db, "SELECT `task`, `material` FROM `p
ractical works` WHERE `id` = '$practical id';");
 $practical = mysqli fetch array($result, MYSQLI ASSOC);
 mysqli close($db);
?>
<!DOCTYPE html>
<html lang="ru">
<head>
 <?php include 'includes/head.php'; ?>
 <link rel="stylesheet" href="css/schedule.css">
 <link rel="stylesheet" href="css/lesson.css">
</head>
<body>
 <?php include 'includes/header.php'; ?>
 <div class="container flex-row">
   <main class="main" id="main">
     <section class="lesson">
       <div class="lesson title">
         <?=$practical['task']?>
         </h2>
       </div>
       <?=$practical['material']?>
```

```
<g\>
        <form action="practical send task.php?practical id=<?=$p</pre>
ractical id?>" method="post">
          <input type="text" name="answer" placeholder="Введите
ответ" required autocomplete="on">
          <input type="submit" value="Отправить">
        </form>
      </section>
    </main>
    <?php include 'includes/sidebar.php'; ?>
  </div>
  <?php include 'includes/footer.php'; ?>
</body>
</html>
Файл lesson_add.php
<?php
  include 'includes/isLogin.php';
  include 'includes/connect.php';
  $subjects id = $ GET['subjects id'];
?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <?php include 'includes/head.php'; ?>
  <link rel="stylesheet" href="css/schedule.css">
  <link rel="stylesheet" href="css/lesson.css">
</head>
<body>
  <?php include 'includes/header.php'; ?>
  <div class="container flex-row">
    <main class="main" id="main">
      <form id="form lesson" class="lesson" action="add.php?add=</pre>
lesson&subjects id=<?=$subjects id?>" method="post" name="lesson
add">
        <div class="lesson title">
          <input form="form lesson" type="text" name="lesson tit</pre>
le" placeholder="Название лекции" required autocomplete="on" aut
ofocus>
          </h2>
        </div>
```

```
<textarea form="form lesson" name="lesson text" cols="</pre>
30" rows="10" placeholder="Текст лекции"></textarea>
        <input form="form lesson" class="add button" type="submi</pre>
t" value="Добавить">
      </form>
    </main>
    <?php include 'includes/sidebar.php'; ?>
  </div>
  <?php include 'includes/footer.php'; ?>
</body>
</html>
Файл practical_add.php
<?php
  include 'includes/isLogin.php';
  include 'includes/connect.php';
  $lesson id = $ GET['lesson id'];
?>
<!DOCTYPE html>
<html lang="ru">
<head>
  <?php include 'includes/head.php'; ?>
  <link rel="stylesheet" href="css/schedule.css">
  <link rel="stylesheet" href="css/lesson.css">
</head>
<body>
  <?php include 'includes/header.php'; ?>
  <div class="container flex-row">
    <main class="main" id="main">
      <form id="form practical" class="lesson" action="add.php?a</pre>
dd=practical&lesson id=<?=$lesson id?>" method="post" name="prac
tical add">
        <div class="lesson title">
          <input form="form practical" type="text" name="practic</pre>
al title" placeholder="Название задания" required autocomplete="
on" autofocus>
```

</h2>

```
<textarea form="form practical" name="practical text"</pre>
cols="30" rows="10" placeholder="Текст задания"></textarea>
        <input form="form practical" type="text" name="practical"</pre>
answer" placeholder="OTBET" required autocomplete="on">
       <input form="form practical" class="add button" type="su</pre>
bmit" value="Добавить">
      </form>
    </main>
    <?php include 'includes/sidebar.php'; ?>
  </div>
  <?php include 'includes/footer.php'; ?>
</body>
</html>
Файл add.php
<?php
  include 'includes/connect.php';
  $profiles id = $ COOKIE['profiles id'];
  $referer = $_SERVER['HTTP REFERER'];
  if ($ GET['add'] == 'discipline') {
    $title = $ POST['title'];
    $result = mysqli query($db, "SELECT * FROM `subjects` WHERE
`name` = '$title' AND `teacher id` = '$profiles id';");
    if (mysqli num rows($result) == 0)
     mysqli query($db, "INSERT INTO `subjects` (name, teacher i
d) VALUES ('$title', '$profiles id')");
  else if ($ GET['add'] == 'lesson') {
    $title = $_POST['lesson_title'];
    $text = $ POST['lesson text'];
    $subjects_id = $_GET['subjects_id'];
    mysqli query($db, "INSERT INTO `lessons` (`subjects id`, `pr
ofiles_id`, `title`, `text`) VALUES ('$subjects id', '$profiles
id', '$title', '$text')");
  else if ($ GET['add'] == 'practical') {
    $task = $ POST['practical title'];
    $material = $ POST['practical text'];
    $answer = $ POST['practical answer'];
    $lesson id = $ GET['lesson id'];
```

```
mysqli query($db, "INSERT INTO `practical works` (`lessons i
d`, `task`, `material`, `answers`) VALUES ('$lesson id', '$task'
, '$material', '$answer')");
 header("Location: $referer");
?>
Файл practical send task.php
<?php
  include 'includes/connect.php';
  $profiles id = $ COOKIE['profiles id'];
  $practical id = $ GET['practical id'];
  $answer = $ POST['answer'];
 header("Location: ". $ SERVER['HTTP REFERER']);
  mysqli query($db, "INSERT INTO `practical messages` (`practica
l works id`, `profiles id`, `text`) VALUES ('$practical id', '$p
rofiles id', '$answer')");
?>
Файл head.php
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-</pre>
scale=1">
<meta http-http-equiv="X-UA-Compatible" content="ie=edge">
<title>Образовательный Веб-сервис</title>
<link href="https://fonts.googleapis.com/css?family=Roboto:400,5</pre>
00&display=swap" rel="stylesheet">
<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="css/style.css">
<script src="js/script.js"></script>
```

Файл header.php

```
<header class="header" id="header">
   <div class="container">
    <div class="header-row">
      <nav>
       class="header-
menu item"><a href="/">Домой</a>
         menu item"><a href="discipline.php">Дисциплины</a>
```

```
class="header-
menu item"><a href="">Обсуждения</a>
          </nav>
        <div class="header-notifications">
          Уведомления
        </div>
        <div class="header-login">
          <a href="authorization.php">Авторизация</a>
        </div>
      </div>
    </div>
  </header>
Файл sidebar.php
<aside class="sidebar" id="sidebar">
  <section class="schedule">
    <article class="schedule-block">
      <header class="schedule-block header">
        <h3>4etbepr</h3>
        <time dateTime="2020-11-09">10.12</time>
      </header>
      <div class="schedule-block-item">
        <h4 class="schedule-block-item name">
          <span class="lesson-type lesson-type-LZ">ЛЗ</span>
          Технологии разработки программного обеспечения
        </h4>
        <span class="schedule-block-</pre>
item teacher">Попова Ольга Борисовна</span>
        <time class="schedule-block-item time">
          <span>09:40</span>
          <span>11:10</span>
        </time>
      </div>
      <div class="schedule-block-item">
        <h4 class="schedule-block-item name">
          <span class="lesson-type lesson-type-LZ">J3</span>
          Технологии разработки программного обеспечения
        </h4>
        <span class="schedule-block-</pre>
item teacher">Попова Ольга Борисовна</span>
        <time class="schedule-block-item time">
          <span>11:20</span>
          <span>12:50</span>
        </time>
      </div>
      <div class="schedule-block-item">
```

```
<h4 class="schedule-block-item name">
          <span class="lesson-type lesson-type-LK">ЛК</span>
          История
        </h4>
        <span class="schedule-block-</pre>
item teacher">Емтыль Зарема Январбиевна</span>
        <time class="schedule-block-item time">
          <span>14:40</span>
          <span>16:10</span>
        </time>
      </div>
      <div class="schedule-block-item">
        <h4 class="schedule-block-item name">
          <span class="lesson-type lesson-type-PZ">∏3</span>
          История
        </h4>
        <span class="schedule-block-</pre>
item teacher">Емтыль Зарема Январбиевна</span>
        <time class="schedule-block-item time">
          <span>16:20</span>
          <span>17:50</span>
        </time>
      </div>
    </article>
    <article class="schedule-block">
      <header class="schedule-block header">
        <h3>Пятница</h3>
        <time dateTime="2020-11-10">11.12</time>
      </header>
      <div class="schedule-block-item">
        <h4 class="schedule-block-item name">
          <span class="lesson-type lesson-type-LK">JIK</span>
          Операционные системы
        </h4>
        <span class="schedule-block-</pre>
item teacher">Дьяченко Роман Александрович</span>
        <time class="schedule-block-item time">
          <span>16:20</span>
          <span>17:50</span>
        </time>
      </div>
    </article>
  </section>
</aside>
```

```
$text-color: #003768;
$first-color: #005BAA;
$first-color-lite: #6099CB;
@font-face {
  font-family: 'Adelle Cyrillic'; /* Имя шрифта */
  src: url(../fonts/AdelleCyrillicRegular.ttf); /* Путь к файлу
со шрифтом */
Файл style.scss
@import "variables.scss";
  box-sizing: border-box;
a {
  text-decoration: none;
  color: inherit;
  &:hover {
    text-decoration: none;
  &.underline {
    position: relative;
    &:before {
      content: '';
      position: absolute;
      bottom: 0;
      left: -2px;
      width: calc(100\% + 4px);
      height: 1px;
      background-color: $first-color;
    }
  }
}
ul {
  list-style: none;
 margin: 0;
 padding: 0;
}
html {
  font-size: 18px;
body {
  overflow-x: hidden;
 background-color: #fff;
```

```
.header, .sidebar, h1, h2, h3, h4, h5, h6 {
  //font-family: $first-font;
// Общие стили
.header, .header-submenu, .sidebar, .sidebar-
menu, .main, .footer {
  position: relative;
 box-shadow: 0px 0px 20px 0px rgba(0,0,0,0.1);
.container {
 width: 1140px;
 margin: 0 auto;
.header {
  z-index: 150;
  margin-bottom: 30px;
}
.main {
  z-index: 50;
 margin-right: 30px;
}
.sidebar {
  z-index: 100;
.footer {
 height: 100px;
 margin-top: 30px;
}
h1 { font-size: 1.5rem; } // 24px
h2 { font-size: 1.25rem; } // 20px
h3 { font-size: 1.125rem; } // 18px
h4 { font-size: 1rem; }
                        // 16px
h5 { font-size: 0.875rem; } // 14px
h6 { font-size: 0.75rem; } // 12px
h1, h2, h3, h4, h5, h6 {
 font-weight: normal;
 margin: 0px;
.flex-row {
  display: flex;
  flex-flow: row nowrap;
}
.bold {
  font-weight: bold;
@media screen and (max-width: 1200px) {
  .container {
    width: 960px;
  }
```

```
@media screen and (max-width: 992px) {
  .container {
    width: 720px;
  }
}
@media screen and (max-width: 768px) {
  .container {
   width: 540px;
 }
}
@media screen and (max-width: 576px) {
  .container {
   width: 100%;
   padding: 0px 10px;
  }
}
@media screen and (max-width: 360px) {
  .container {
   // padding: 0;
 }
}
.header{
  font-family: "Roboto", sans-serif;
  font-size: 1.125rem;
  &-row {
   display: flex;
    align-items: center;
  &-menu {
   display: flex;
    padding: 10px 0px 15px 0px;
    & item {
      position: relative;
      margin-right: 20px;
      background-color: transparent;
      letter-spacing: 1px;
      \& > a \{
        display: inline-block;
        padding: 12px 20px;
        position: relative;
        z-index: 10;
      &:before {
        content: '';
        position: absolute;
        left: 0;
        bottom: 0;
```

```
z-index: 0;
        width: 100%;
        height: 2px;
        background-color: $first-color;
        transition: ease-out 0.2s;
      }
      &:hover {
        color: white;
        &:before {
          height: 100%;
    }
  }
  &-notifications {
   margin-left: auto;
  &-login {
   margin-left: 15px;
  }
}
.main {
 width: 100%;
 //min-height: 100%;
 padding: 20px;
}
.sidebar {
 padding: 6px 12px 12px 12px;
 font-family: Roboto, sans-serif;
.add button {
 display: inline-flex;
  justify-content: center;
 align-items: center;
 background-color: $first-color;
 color: white;
 border: none;
 padding: 4px 12px;
 cursor: pointer;
}
input {
  font-
family: Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helveti
ca Neue', sans-serif;
 border: 2px solid $first-color;
 padding: 6px 10px;
  &:active {
   border: 2px solid $first-color;
```

```
}
input[type=submit] {
 background-color: $first-color;
 padding: 6px 16px;
 color: white;
 cursor: pointer;
}
Файл home.scss
@import "variables.scss";
.person {
 display: flex;
  flex-flow: row nowrap;
 margin-right: 30px;
  & photo {
   max-width: 160px;
    margin-right: 15px;
    & > img {
      width: 100%;
    }
  }
  &-content {
    & name {
      margin-bottom: 10px;
    }
    & > div {
     margin-bottom: 6px;
    }
  }
}
.tasks {
 margin-top: 15px;
  & header {
    & > span {
      display: inline-block;
      position: relative;
      padding: 10px 15px;
      &:before {
        content: '';
        position: absolute;
        left: 0;
        bottom: 6px;
        z-index: 0;
        width: 100%;
```

```
height: 2px;
        background-color: $first-color;
    }
  &-list {
    & item {
      margin-top: 8px;
      & > a > .abbreviation {
       padding-right: 6px;
        border-right: 2px solid $first-color;
      }
    }
  }
Файл schedule.scss
@import "variables.scss";
.schedule {
 width: 400px;
 margin-left: auto;
  &-block {
   padding: 4px 14px 10px 14px;
    & header {
      width: 100%;
      display: flex;
      align-items: center;
      justify-content: space-between;
      padding: 8px 12px;
      margin-bottom: 12px;
     border-bottom: 2px solid $first-color;
    }
    &-item {
      margin-top: 10px;
      display: grid;
      grid:
        "schedule-item__name schedule-item__time"
        "schedule-item teacher schedule-item time"
        / auto 50px;
      & name {
        grid-area: schedule-item name;
        line-height: 1.25rem;
      & teacher {
        grid-area: schedule-item teacher;
```

font-size: 0.875rem;

```
margin-top: 4px;
      & time {
        position: relative;;
        grid-area: schedule-item time;
        display: flex;
        flex-flow: column;
        justify-content: center;
        align-items: center;
        & > span:nth-child(1) {
          margin-bottom: 6px;
        &::before {
          content: '';
          position: absolute;
          top: calc(50% - 1px);
          width: 80%;
          height: 1px;
          background-color: $first-color;
      }
    }
  }
  .lesson-type {
   padding: 2px 6px;
    &-LK {
      background-color: #FFEC8B;
    }
    \&-LZ {
      background-color: #AB82FF;
    }
    &-PZ {
      background-color: #87CEFF;
    }
}
Файл discipline.scss
@import "variables.scss";
.search {
  &-field {
    font-family: Roboto, sans-serif;
    display: flex;
    justify-content: space-around;
    border-bottom: 2px solid $first-color;
```

```
padding-bottom: 10px;
  &-results {
    margin-top: 10px;
    & item {
      margin-bottom: 8px;
    }
  }
}
.discipline add {
  font-family: Roboto, sans-serif;
  display: flex;
  justify-content: space-around;
  padding-bottom: 10px;
  & > input {
    border: 2px solid $first-color;
    padding: 6px 10px;
    &:active {
      border: 2px solid $first-color;
    }
  & > input[type=submit] {
    background-color: $first-color;
    padding: 6px 16px;
    color: white;
    cursor: pointer;
  }
}
Файл lesson.scss
@import "variables.scss";
.lesson {
  & title {
    \& > h2  {
      display: inline-block;
      padding: 4px 12px;
      border-bottom: 2px solid $first-color;
    }
  & text {
    text-align: justify;
    & > textarea {
      width: 100%;
    }
  }
}
```

```
.practicals {
  & title {
    display: flex;
    justify-content: space-between;
    \& > h2  {
      padding: 4px 12px;
      border-bottom: 2px solid $first-color;
    }
  }
  &-list {
    margin-top: 10px;
    & item {
     margin-top: 8px;
   }
  }
}
.add button {
  font-
family: Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helveti
ca Neue', sans-serif;
 display: inline-flex;
 justify-content: center;
  align-items: center;
 background-color: $first-color;
 color: white;
 border: none;
 padding: 10px 17px;
 cursor: pointer;
}
Файл login.scss
@import "variables.scss";
.connect {
  &-wrap {
    width: 100vw;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
  display: flex;
  flex-flow: column;
  width: 440px;
```

```
font-
family: "Roboto", 'Segoe UI', Tahoma, Geneva, Verdana, sans-
serif;
  font-size: 1rem;
 background-color: #fff;
  & title, & form {
    padding: 20px 20px 16px 20px;
  & title {
    color: #fff;
    background-color: $first-color;
    font-family: inherit;
    font-weight: 500;
    text-align: center;
    margin: 0;
  }
  & form {
   display: flex;
    flex-flow: column;
    align-items: center;
    border: 2px solid $first-color;
    & > input {
      width: 100%;
      border-radius: 10px;
    & > input[type="text"],
    & > input[type="password"] {
      height: 40px;
      padding: 3px 7px;
      margin-bottom: 10px;
      background: #ffffff !important;
      border: 1px solid #ccccc;
      outline: none;
      &:focus{
        box-
shadow: Opx Opx 6px 1px transparentize ($color: $first-
color, $amount: 0.75);
        border-color: transparentize ($color: $first-
color, $amount: 0.75);
      }
      &.error {
        border: 1px solid red;
        &:focus {
          box-
shadow: Opx Opx 6px 1px transparentize ($color: red, $amount: 0.7
5);
        }
      }
```

```
& > input[type="submit"] {
     cursor: pointer;
      height: 46px;
      border-width: 0;
      margin-top: 10px;
      color: #fff;
      background-color: transparentize ($color: $first-
color, $amount: 0.5) !important;
      //opacity: 0.2;
      &:hover{
       background-color: #113e66 !important;
   }
  }
  & else {
   display: inline-block;
   margin-top: 8px;
   font-size: 0.875rem;
   color: $text-color;
   text-decoration: underline;
  }
}
.message {
 position: absolute;
 top: 50%;
  left: 50%;
 max-width: 400px;
 transform: translate(-50%, -50%);
 padding: 15px 20px;
 text-align: center;
 background-color: white;
```

Приложение Б

