# СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 1](#_Toc119583194)

[Основные функции сайта 3](#_Toc119583195)

[Определение группы пользователей 4](#_Toc119583196)

[Разработка дизайна 5](#_Toc119583197)

[Проектирование базы данных 7](#_Toc119583198)

[Введение в Yii 8](#_Toc119583200)

[Разработка регистрации и авторизации 9](#_Toc119583205)

[Формирования отчета 15](#_Toc119583215)

[Слияния посредством GitHub 18](#_Toc119583223)

[ЗАКЛЮЧЕНИЕ 20](#_Toc119583224)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#_Toc119583225)

# ВВЕДЕНИЕ

Сайты хранят в себе массу полезной информации, предоставляют актуальные услуги, они находятся в открытом доступе для любого пользователя, а поэтому так широко пользуются спросом на сегодняшний день.

Каждый сайт имеет свою собственную целевую аудиторию, которая объединяется одними интересами и находится в поиске определённой информации, а если эта информация собрана в одном месте с удобным, интуитивно понятным минималистичным интерфейсом – такой сайт будет цениться гораздо больше.

В данной учебной практике был создан сайт с использованием фреймворка Yii. В ходе разработки был реализован интерфейс сайта, авторизация, регистрация пользователя и т.д. Основная работа с дизайном была выполнена в таких программах как Figma, Adobe Photoshop. Редактором кода был выбран PhpStorm.

Для выполнения данной работы необходимо выполнить следующие задачи:

* Разработка и проектирование базы данных;
* Разработка интерфейса;
* Реализация вывода данных с помощью Yii;
* Написание кода для грамотной авторизации и регистрации;
* Слияние отдельных модулей проекта с помощью GitHub;

Основные функции сайта

Сайт Сollege Сore реализует множество функций, каждая из них направлена на улучшение жизни студентов и преподавателей.

Личный кабинет содержит в себе календарь, даты в нем разделены цветами на выходные дни, учебные и СДО, краткую сводку погоды, расписание пар на сегодня, расписание звонков, виджет с датой сдачи самого ближайшего задания, расположение корпусов, диаграмму, отражающую в процентах количество оставшихся дней учебного года.

Страница отчетов состоит из списка работ, тем, сроков сдачи и статуса в котором находится данная работа. Открыв информацию о работе, можно сформировать шаблон отчета для этой работы, основанный на данных пользователя. Так же, здесь присутствует фильтр по статусу работ.

Определение группы пользователей

Главная цель College Core – привлечение конкретной аудитории, которая будет заинтересована в функционале сайта. Важно понимать, кто будет целевой аудитории. Это позволит грамотно выстроить стратегию, чтобы развивать сайт в правильном направлении. Именно такой подход позволит учесть потребности потенциальных посетителей, а значит сделать сайт востребованным и популярным

В начале разработки проекта был произведен анализ аудитории, затем утверждена концепция. Выявлено, что пользователями College Core являются ученики, поэтому сайт должен содержать всю информацию об учебном процессе и факторах, что могут оказывать на него влияния извне, например-погодные условия.

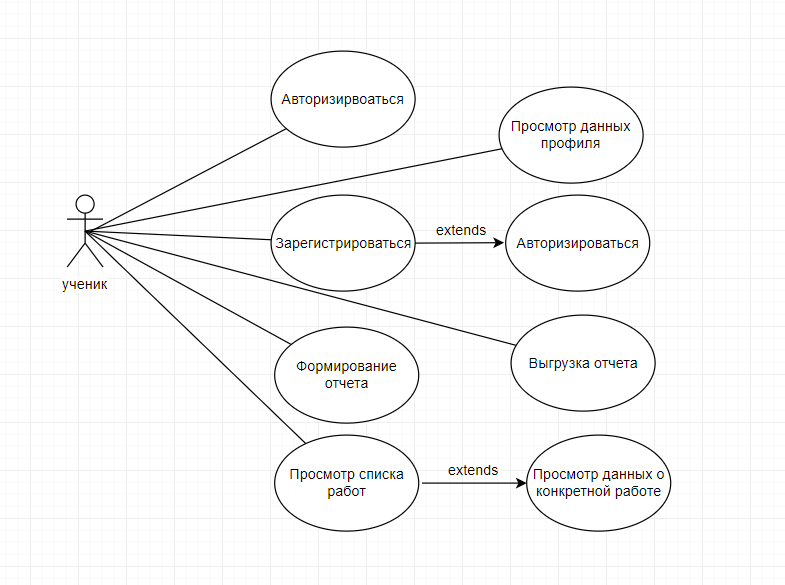


Рисунок 1-Use-Case диаграмма

Разработка дизайна

Основываясь на понимании того, кто является целевой аудиторией, была утверждена концепция. При создании сайтов один из важнейших этапов – разработка внешнего вида интерфейса, важно не только содержание и наполнение, но и качественный дизайн, который дополнит и подчеркнет структуру страницы.

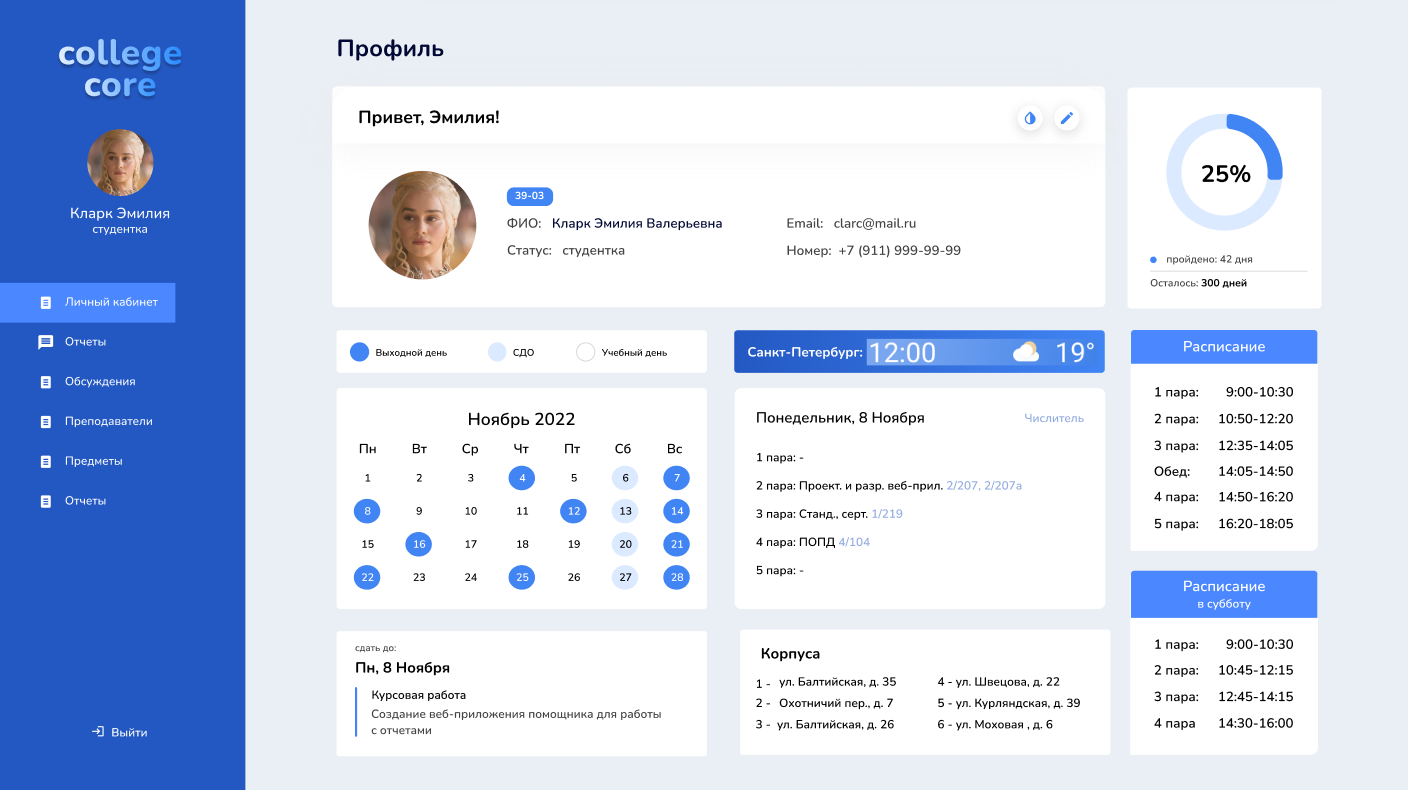
В процессе разработки дизайна созданы и отброшены множества вариантов, было решено остановиться на синем цвете, и стилизации блоков в виде виджетов. Считается, что классический синий — консервативный цвет, он успокаивает и вызывает доверие, символизирует ответственность и ум. У человека возникают ассоциации с природой — небом, водой, а также прохладой, — а значит, он расслабляется. Этот цвет в равной мере считают любимым и мужчины, и женщины.  
  


Рисунок 2 - профиль пользователя

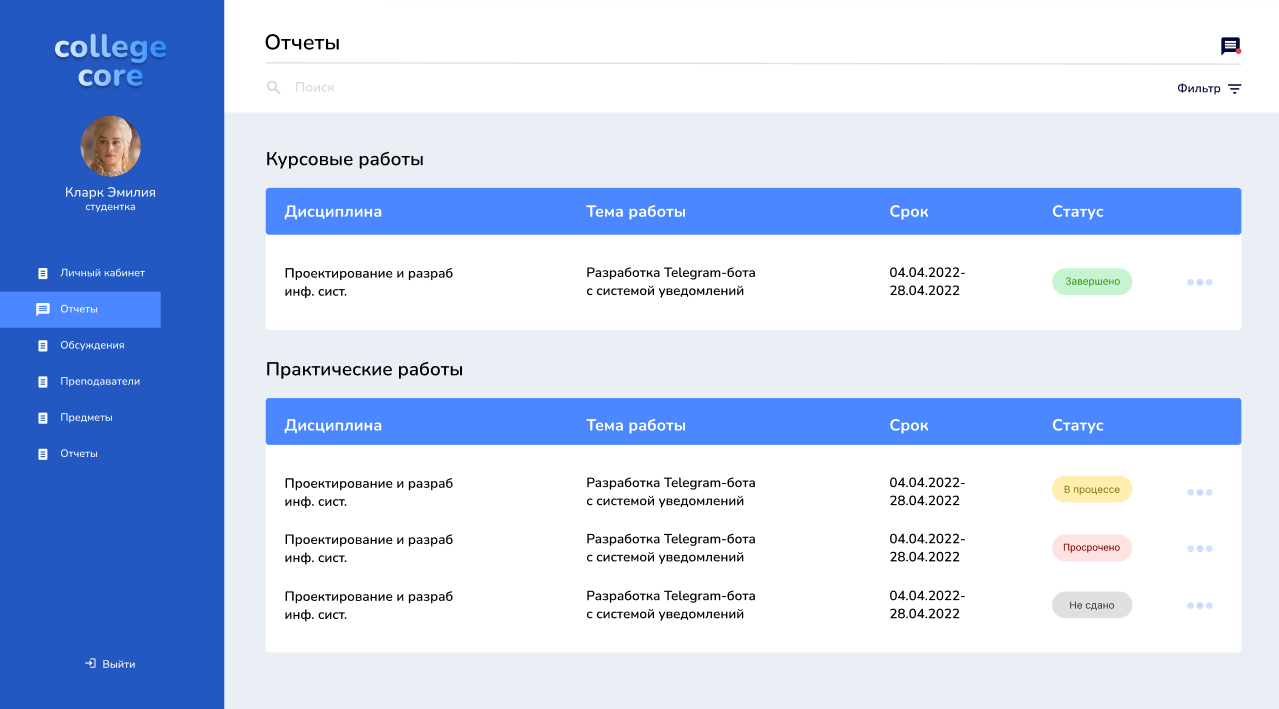


Рисунок 3.1 - страница с отчетами

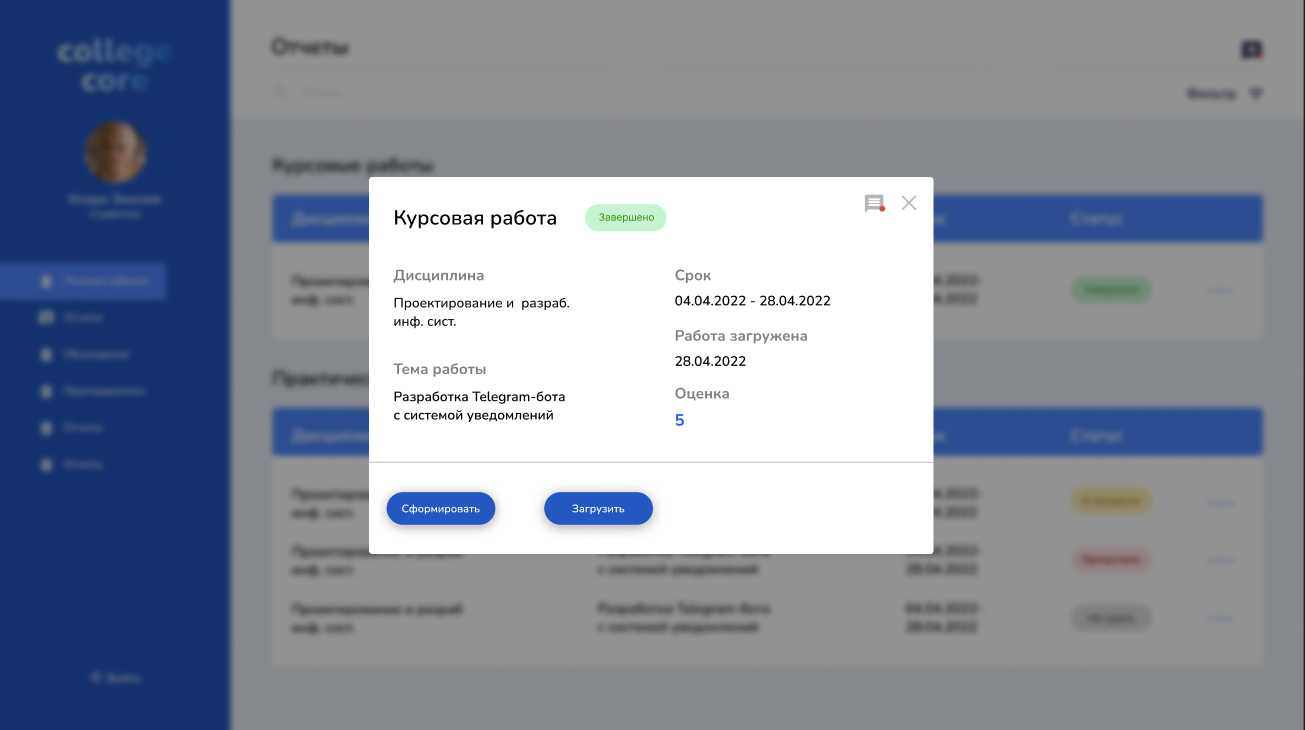


Рисунок 3.2- данные о работе

Проектирование базы данных

Для College Core, необходимо создать информационную систему, позволяющую хранить, вводить, редактировать и удалять данные.

Для проектирования базы данных был применен метод сущность-связь. Сущность-связь – это модель предметной области, которая позволяет моделировать объекты ПО и их взаимоотношения. В основе модели лежат три конструктивных элемента: сущность, атрибут и связь. Сущность – это некоторая абстрактная реально существующего объекта, процесса или явления, о котором необходимо хранить информацию в системе. Атрибут – это характеристика сущности. Связи – это средства, с помощью которых представляются отношения между сущностями.

В результате анализа предметной области проекта была разработана база данных.

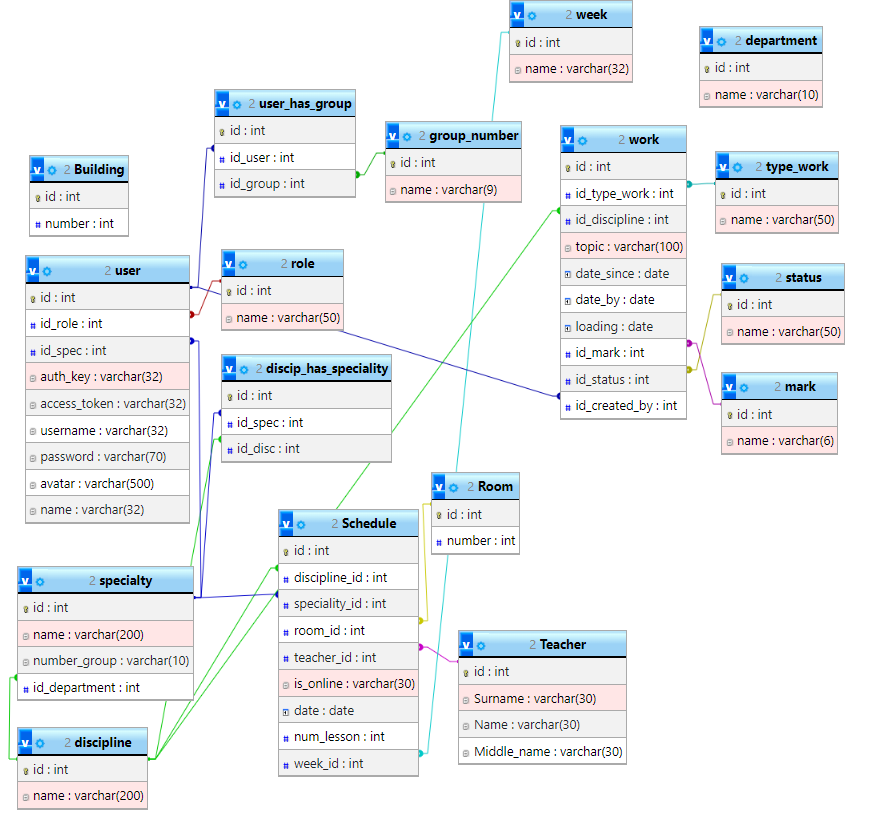


Рисунок 5 - база данных проектов

Введение в Yii

Для реализации проекта был выбран Yii- это объектно-ориентированный фреймворк, написанный на PHP. Так как PHP один из самых популярных языков программирования на сегодняшний день, то и фреймворков, написанных на нем, немало: Laravel, CakePHP, Symfony, CodeIgniter и многие другие. Yii можно смело назвать одним из лидирующих по популярности фреймворком.

Преимущества Yii - это высокая скорость работы и хорошая поддержка OОП. Yii включает в себя большое количество библиотек. Благодаря им можно легко создать веб-приложение, которое будет отвечать всем современным стандартам. Встроенные методы позволяют значительно сокращать количество кода.

Важно, что Yii не только распространяется бесплатно, но и выходит под лицензией BSD. То есть его можно использовать в коммерческих целях, встраивать в проприетарные продукты.

Yii имеет хорошую документацию на русском языке.Фреймворк Yii известен своим огромным сообществом, в том числе на русском языке. Большие сообщества разработчиков - это возможность оперативно получить помощь и обсудить важные темы.

Так же, ii имеет гибкий механизм кэширования и поддерживает несколько хранилищ для кэширования переменных, участков кода и так далее.

Yii - фреймворк имеет достаточно низкий порог вхождения, однако, как и любой другой инструмент, он требует некоторого времени для изучения, чтобы потом можно было использовать его максимально эффективно и удобно.

Разработка регистрации и авторизации

Реализация регистрации, как и весь проект был построена на взаимодействии модели контроллера и вывода данных.

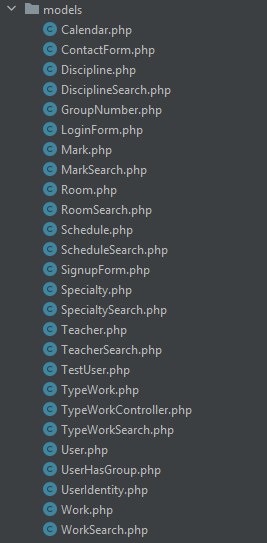


Рисунок 6-модели проекта

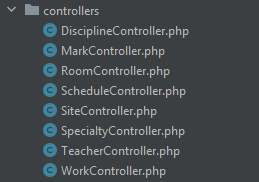


Рисунок 7-контроллеры проекта

Так как, college core создан исключительно студентов и преподавателей колледжа, регистрации доступна только при наличии учетной записи в СДО.

Листинг 1.1 – Проверка наличия аккаунта в СДО

$model = new LoginForm();  
 if ($model->load(Yii::$app->request->post())) {  
 $ch = curl\_init();  
// $model->load(Yii::$app->request->post()) && $model->login();  
 curl\_setopt($ch, CURLOPT\_URL, 'https://portal.petrocollege.ru/');  
 curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);  
 curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, 'GET');  
 curl\_setopt($ch, CURLOPT\_HTTPAUTH, CURLAUTH\_NTLM);  
 curl\_setopt($ch, CURLOPT\_UNRESTRICTED\_AUTH, true);  
  
 curl\_setopt($ch, CURLOPT\_USERPWD, ($model['username'] . ':' . $model['password']));  
//  
 $result = curl\_exec($ch);  
 if (curl\_errno($ch)) {  
 echo 'Error:' . curl\_error($ch);  
 } elseif ($result == "") {  
 echo '<script>alert("Введите логин/пароль от СДО")</script>';  
 } else {  
 echo '<script>alert("Добро пожаловать")</script>';  
 $user\_sdo = new UserIdentity();

Функция реализована с помощью библиотеки cURL .

cURL PHP – это библиотека предназначенная для получения и передачи данных через такие протоколы, как HTTP, FTP, HTTPS. Библиотека используется для получения данных в виде XML, JSON и непосредственно в HTML, парсинга, загрузки и передачи файлов и т.д.

В Листинг 1.1 так же видно работу с моделью LoginForm().

Если человек извне, не принадлежащий к колледжу попытаемся авторизоваться или зарегистрироваться он получит сообщение "Введите логин/пароль от СДО".

После заполнения пользователем формы авторизации или регистрации, начинается проверка наличия пользователя в базе данных.

Листинг 1.2 – Проверка наличия в БД

$user\_sdo = new UserIdentity();  
//  
 if (!UserIdentity::findByUsername($model['username'])) {  
 $user\_sdo->username = $model['username'];  
 $user\_sdo->password = $model->password;  
 $user\_sdo->name = $model->name;  
 $user\_sdo->save();  
 $group = new UserHasGroup();  
 $group->id\_user = $user\_sdo->id;  
 $group->id\_group = $model->group\_id;  
 $group->save();  
 if ($model->load(Yii::$app->request->post()) && $model->login()) {  
 return $this->redirect('site/profile');  
 }  
 }  
 if ($model->load(Yii::$app->request->post()) && $model->login()) {  
 return $this->redirect('site/profile');  
 }  
 }

Если аккаунт пользователя существует на СДО, но он отсутствует в БД Core, создается новый пользователь со своим именем, id, логином, паролем. Так же в таблице со связью многие ко многим появляется запись о пользователе и номере его группы.

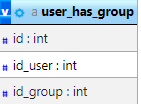


Рисунок 8-таблица с ключами пользователя и группы

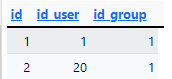


Рисунок 9-вид таблицы

Данные ограничиваются правилами в моделях.

Листинг 1.3 – Проверка данных модель User

public function rules()  
{  
 return [  
 [['id\_role', 'id\_spec'], 'integer'],  
 [['username',], 'required'],  
 [['auth\_key', 'access\_token', 'username'], 'string', 'max' => 32],  
 [['password'], 'string', 'max' => 70],  
 [['avatar'], 'string', 'max' => 500],  
 [['name'], 'string', 'max' => 50],  
 [['id\_role'], 'exist', 'skipOnError' => true, 'targetClass' => Role::class, 'targetAttribute' => ['id\_role' => 'id']],  
 [['id\_spec'], 'exist', 'skipOnError' => true, 'targetClass' => Specialty::class, 'targetAttribute' => ['id\_spec' => 'id']],  
 ];  
}

Листинг 1.4 – Проверка данных модель Login Form

public function rules()  
{  
 return [  
 // username and password are both required  
 [['username', 'password'], 'required','message' => ' Поле не может быть пустым'],  
 'name' => [['name'], 'string', 'max' => 60],  
 ['name', 'match', 'pattern' => '/^[a-z]+$/', 'message' => 'Допустимы только буквы'],  
 [['group\_id'],'integer'],  
 // rememberMe must be a boolean value  
 ['rememberMe', 'boolean'],  
 // password is validated by validatePassword()  
 ['password', 'validatePassword'],  
 ];  
}

В ходе работы была реализована регистрация отдельно от авторизации.

Листинг 1.5 – Регистрация

public function actionSignup()  
 {  
 if (!Yii::$app->user->isGuest) {  
 return $this->goHome();  
 }  
 $model = new SignupForm();  
 if ($model->load(Yii::$app->request->post()) && $model->validate()) {  
 $user = new UserIdentity();  
  
  
 $user->username = $model->username;  
 $user->password = $model->password;  
 $user->save();  
 $group = new UserHasGroup();  
 $group->id\_user = $user->id;  
 $group->id\_group = $model->group\_id;  
 var\_dump($group);  
 $group->save();  
 VarDumper::dump($user->errors, 10, true);  
  
// $user ->password = Yii::$app->security->generatePasswordHash($model->password);  
 if ($user->save()) {  
 Yii::$app->user->login($user);  
// var\_dump( $user->password );  
//  
 return $this->goHome();  
  
 }  
  
  
 }  
  
 return $this->render('signup', ['model' => $model]);  
  
 }

Однако, чуть позже было решено совместить в один метод авторизацию и регистрацию, отдельная страница регистрации была удалена. Пользователь выбирает хочет он авторизоваться или зарегистрироваться, на основе этого открываются поля для заполнения. Проверка существования записи в БД всегда проходит на случай человеческого фактора.

Формирования отчета

Отчет состоит из дневника, написанного студентом и стандартных страниц, содержащих данные о работе: место прохождения практики, ФИО преподавателя, ФИО ученика, номер телефона куратора и т.д. Написания всех этих пунктов с поддержанием верной стилистики, это задача, которую можно автоматизировать.

Для реализации этой идеи был выбран PHP Word. Библиотека имеет множество различных функций: добавлять новые разделы в документ, ставить значения шрифтов, цветов по умолчанию и т.д.

Для College Core требуется поместить данные из формы в отчет сохранив шрифт, его размер, цвет.

Листинг 1.6 –Подключение и объявление переменных  
$document = new \PhpOffice\PhpWord\TemplateProcessor('./input.docx');  
$uploadDir = \_\_DIR\_\_;  
$outputFile = 'review\_ful.docx';  
$name = $\_POST['name'];  
$group= $\_POST['group'];  
$place = $\_POST['place'];  
$until = $\_POST['until'];  
$since = $\_POST['since'];  
$teacher = $\_POST['teacher'];  
$work = $\_POST['work'];  
$tel = $\_POST['tel'];

В демо версии приложения данные вводятся в форму, при входе в аккаунт такие данные как ФИО, курс пользователя форма будет получать автоматически из БД.

Затем, данные передаются документу.

Листинг 1.7–Заполнение файла Word  
$document->setValue('name',$name);  
$document->setValue('tel',$tel);  
$document->setValue('group',$group);  
$document->setValue('place',$place);  
$document->setValue('until',$until);  
$document->setValue('since',$since);  
$document->setValue('teacher',$teacher);  
$document->setValue('work',$work);  
$document->saveAs($outputFile);  
  
$fileup = $outputFile;

Как только отчет формируется начинается автоматическое скачивание файла в папку загрузки на компьютер.

Листинг 1.8–Скачивание файла

header("Content-Type: application/octet-stream");  
  
header("Accept-Ranges: bytes");  
  
header("Content-Length: ".filesize($fileup));  
  
header("Content-Disposition: attachment; filename=".$fileup);

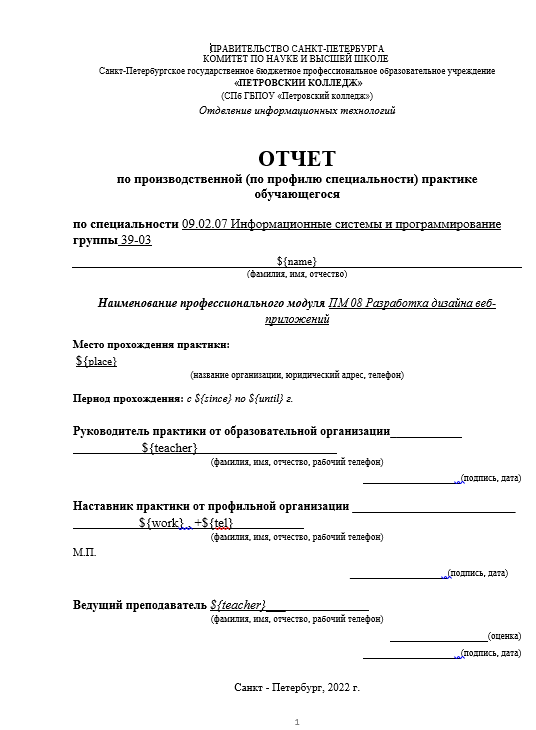


Рисунок 10-отображение файла шаблона

Слияния посредством GitHub

Модули программы с различных веток требуют слияния. В Git есть два вида слияния: перемотка (fast-forward merge) и «истинное» слияние (true merge).

#### «Истинное» слияние (true merge). Случай, при котором разработчик отклоняется от ветки master, чтобы внести улучшения.

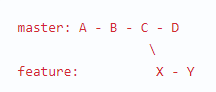


Рисунок 11-история коммитов

После выполнения git merge feature на ветке master, история коммитов будет выглядеть иначе.

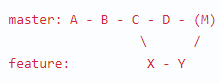


Рисунок 12-cлияние

Это наиболее частый паттерн слияния. В данном случае в ветке master создается новый коммит (M), который будет ссылаться на двух родителей: коммит D и коммит Y; а указатель master установится на коммит (M). Таким образом Git будет понимать, какие изменения соответствуют коммиту (M) и какой коммит последний в ветке master. Обычно коммит слияния делается с сообщением вроде «Merge branch 'feature'».

Другим способ слияния является Перемотка (fast-forward merge).

В ветке master нет коммитов после ответвления.

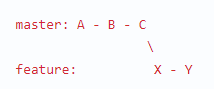


Рисунок 13-история коммитов

В этом случае отсутствует коммит слияния, указатель (ветка) master просто устанавливается на коммит Y, туда же указывает и ветка feature.



Рисунок 14-процесс слияния

Чтобы предотвратить перемотку, можно использовать параметр --no-ff.  
В случае, если мы выполним git merge feature --no-ff -m '(M)'.

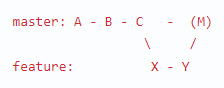


Рисунок 15-при использовании параметра --no-ff

После слияния все модули сайта работали верно, однако после принятого решения не использовать bootstrap, потребовалось написать дополнительной код для отображения.

# ЗАКЛЮЧЕНИЕ

В ходе учебной практики, были проведены: выбор средств разработки, создание схемы БД структуры проекта, разработка интерфейса и реализация самого приложения, изучение Yii, улучшение навыков программирования на PHP.В результате проделанной работы были достигнуты все поставленные цели.

По итогу работы был создан сайт для пользователей СДО, позволяющий авторизоваться, регистрироваться, формировать отчеты, просматривать профиль и статус работ.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Слияние // GitHub URL: https://git-scm.com/book/ru/v2/ (дата обращения: 14.11.2022).
2. Формирование отчета // PHPWord URL: https://www.w-lab.ru/baza-znanii/php-mysql/sozdanie-word-docx-dokumentov-na-php/ (дата обращения: 11.11.2022).
3. Программирование // cURL URL: https://snipp.ru/php/curl (дата обращения: 07.11.2022).
4. Пользовательский интерфейс // UI URL: https://ux-journal.ru/chto-takoe-polzovatelskij-interfejs.html#ui-kit
5. Оптимизация изображений для web // КОМЬЮНИТИ URL: https://timeweb.com/ru/community/articles/optimizaciya-izobrazheniy-dlya-web-1 (дата обращения: 11.11.2022).
6. ТЕОРИЯ ЦВЕТОВОЙ ГАРМОНИИ // photodrum URL: https://photodrum.com/teoriya-tsvetovoj-garmonii/ (дата обращения: 11.11.2022).
7. Полное руководство по Yii 2.0 // Yii URL: https://www.yiiframework.com/doc/guide/2.0/ru (дата обращения: 10.11.2022).
8. Полное руководство по Yii 2.0 // Yii URL: https://www.yiiframework.com/doc/guide/2.0/ru/db-active-record (дата обращения: 5.11.2022).