МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №34

КУРСОВАЯ РАБОТА (ПРОЕКТ)   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ст. преподаватель |  |  |  | К.А. Жиданов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ  “TopFood – сайт с рецептами блюд” |
| по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3135 |  |  |  | Н.В. Ярошенко |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

Оглавление

[**I. Введение** 3](#_Toc91294687)

[**II. Ход работы** 4](#_Toc91294688)

[**1. Выбрать систему управления проектами** 4](#_Toc91294689)

[**2) Выбрать систему контроля версий** 6](#_Toc91294690)

[**3) Выбрать хостинг** 7](#_Toc91294691)

[**4) Написать техническое задание** 7](#_Toc91294692)

[**5) Процесс работы по спринтам** 8](#_Toc91294693)

[**Вывод:** 10](#_Toc91294694)

# **I. Введение**

**Цель** данной работы: изучить технологии разработки информационной системы. Разработать онлайн сервис помощи студентам.

**Актуальность** курсовой работы заключается в отсутствии структурированной и полной информации по предметам, которые изучаются в университете.

В данной работе используется **Agile-разработка** (гибкая методология разработки) - это серия методов разработки ПО, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля. В Agile-разработке реализовано цикличное развитие. Здесь, в отличие от каскадного проектирования, можно совершить шаг назад для выявления и решения проблем. Цель Agile-разработки - как можно быстрее получить Minimum Valuable Product (минимально значимый продукт). Основные ценности Agile таковы:

1. люди и взаимодействие важнее процессов и инструментов;
2. работающий продукт важнее исчерпывающей документации;
3. сотрудничество с заказчиком важнее согласования условий контракта;

готовность к изменениям важнее следования первоначальному плану.

**Scrum(скрам)** - один из методов Agile-разработки, в котором поставка продукта осуществляется в рамках серии спринтов – коротких (1-2 недели) итераций с одинаковой длительностью. Так большой сложный проект разбивается на небольшие задачи.

1. Планирование спринта - короткой итерации (1-2 недели). Результат - список задач, которые нужно решить сейчас (бэклог спринта);
2. Определить критерии проверки для unit-теста - небольшого фрагмента кода продукта, написанного и поддерживаемого разработчиками в группе;
3. По завершении спринта провести обзор итогов спринта;
4. Осуществить ретроспективу спринта: выполнена ли цель? какие проблемы возникли? как их устранить?

По завершении этого цикла планируется следующий спринт.

По сути, Scrum - работа с некоторыми инструментами по спринтам в группе от 3-х до 10-ти человек.

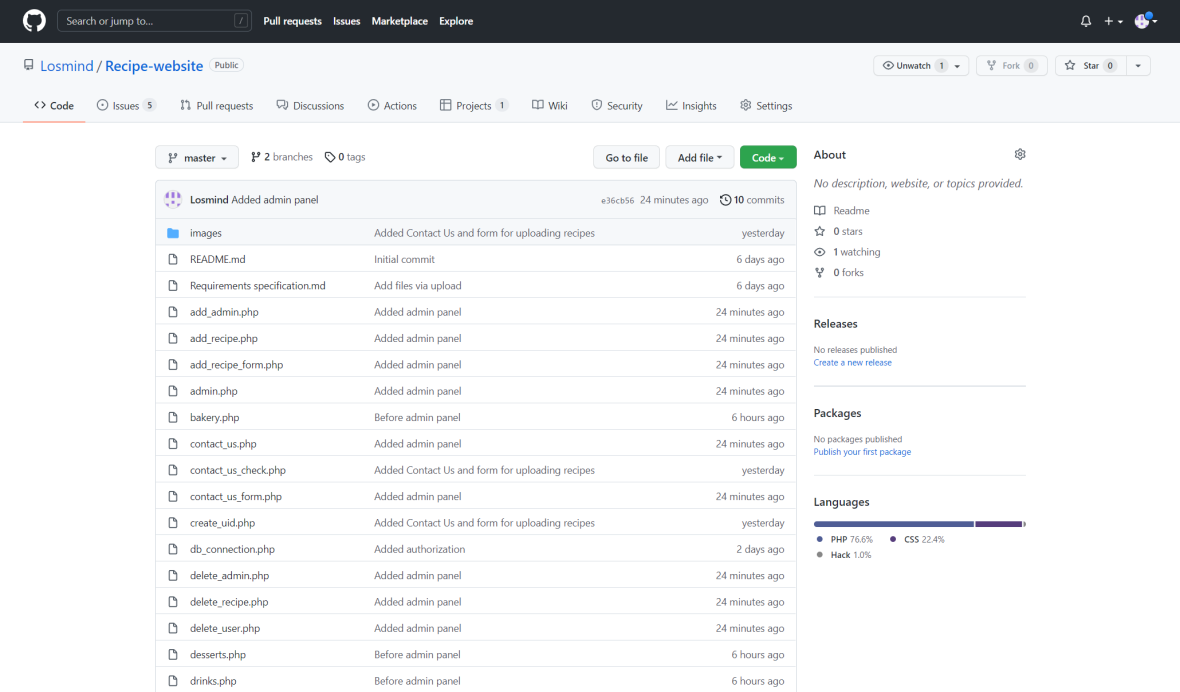
**Канбан** - это метод управления разработкой по принципу "точно в срок" и способствующий равномерным распределением нагрузки между работниками. При данном подходе весь процесс разработки прозрачен для всех членов команды. Задачи по мере поступления заносятся в отдельный список, откуда каждый разработчик может извлечь требуемую задачу. Задачи наглядно расположены в соответствии со статусом, они не изолированы друг от друга и взаимозаменяемы. Типичная канбан-доска делится на 3 колонки:

1. Задачи, которые необходимо выполнить (to-do-лист);
2. Задачи, которые в текущий момент находятся в работе;
3. Задачи, которые уже выполнены и находятся на доске исключительно для отслеживания прогресса.

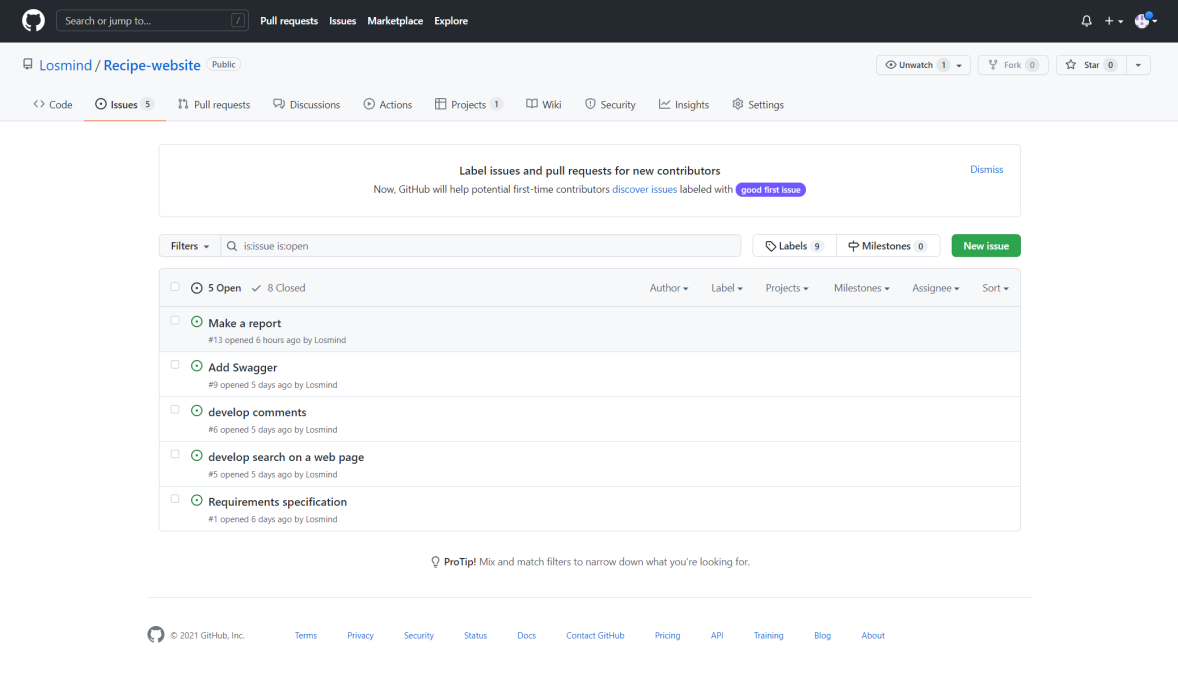
**II. Ход работы**

## **1. Выбрать систему управления проектами**

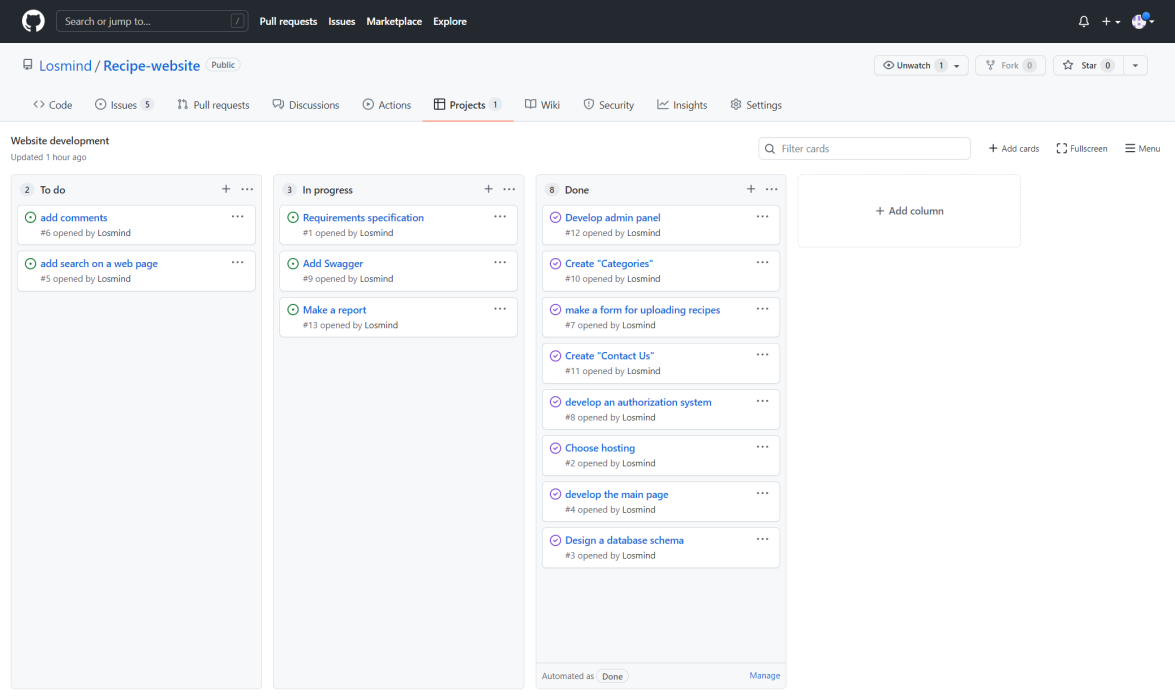
GitHub – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub. GitHub включает в свой функционал канбан и скрам-доски. Этот функционал дополняются массой вспомогательных механизмов, которые добавлялись в приложение исключительно с целью упростить создание новых приложений, добавить в них функции, исправить ошибки и т.п. Также GitHub позволяет создавать ветки разработки проектов.

Интерфейс GitHub делится на несколько ключевых вкладок. Во вкладке "**<>**Code" хранятся весь код нашего проекта, который можно просматривать или редактировать. Это основное рабочее пространство. Здесь же присутствует режим отслеживания релизов продукта, взглянуть на все активные спринты, проанализировать отчеты о проделанной работе.

Раздел Issues включает в себя все текущие задачи (задачи в оригинале называются Issues - "проблемы"). Issue – единица информации. В неё закладывается либо какая-то функция, которую нужно реализовать, либо ошибка в программе, которую необходимо исправить.

Issues – это составные части проекта и спринта. Именно список задач формирует рабочий процесс. Поэтому он и состоит из создания задач, наблюдения за ними, выполнения, анализа, дополнения, изменения и т.п.

Во вкладке **"Projects"** отображены все наши проекты. В этом разделе можно управлять проектом (Редактировать/закрыть).

В самом проекте расположена scrum-доска – это канбан-доска для разработчиков, в ней можно добавить новую задачу или спринт (перетащив в него задачи), завершить активный спринт. Только что добавленные задачи можно добавлять в новый или активный спринт.

Одно из преимуществ GitHub – возможность тесно интегрировать ее с другими продуктами, например, с программой контроля версий Git.  Так появляется возможность наблюдать не только за прогрессом как за набором меняющихся активных задач, но и смотреть на реальные изменения в коде.

## **2) Выбрать систему контроля версий**

**Система контроля версий (Version Control System, VCS)** позволяет хранить несколько версий одного и того же кода, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение. **Система контроля версий Git** фиксирует редактирования кода - добавление или удаление строк. Git позволяет сделать diff - инструмент, показывающий правки и сохраняющий файлы и изменения в нем.

Каждое изменение или сохранение хранится в Git и называется коммит(commit). Репозиторий - это место хранения всех коммитов.

Ветка в Git - это подвижный указатель на один из коммитов. Обычно ветка указывает на последний коммит в цепочке коммитов. Ветка берет свое начало от последнего коммита, который находится в той ветке, в которой находится разработчик. В Git ветка по умолчанию - это master.

Такой интерфейс реализован на веб-сервере GitHub, основанном на системе контроля версий Git.

## **3) Выбрать хостинг**

Для размещения сайта необходимо выбрать хостинг - физическое размещение веб-сервера. Такую услугу предоставляет **Oracle cloud**. После регистрации на сайте **Oracle cloud** пользователь может выбрать бесплатный тариф (trial) на неограниченное время. Так же там есть множество платных тарифов. Все они предоставляют разные объёмы файлового пространства и возможность разместить разное количество сайтов.

Доменное имя — символьное имя, служащее для идентификации областей — единиц административной автономии в сети Интернет — в составе вышестоящей по иерархии такой области. Каждая из таких областей называется доменом. Домен будет зарегистрирован в FREENOM –поставщик доменов. Здесь можно приобрести домен на год. После регистрации собственного домена пользователь может управлять им.

## **4) Написать техническое задание**

Сайт с рецептами блюд:

Необходимо предложить пользователю рецепты с подробным описанием и фотографиями, полезные в готовке.

На целевой странице должны находиться:

* Категории рецептов
* ссылки разделы сайта с переходом на страницы сайта
* форму форма авторизации

Другие страницы сайта будут разделены на вкладки:

1. “Связь с нами” – содержит информацию об администрации сайта и форма связи с администрацией;
2. “Добавить рецепт” – включает в себя форму для добавления роцепта;
3. “Вход/Регистрация” – являются модальными окнами, в которых пользователь в зависимости от выбора зарегистрироваться или авторизоваться;
4. Если пользователь уже авторизовался у него появится кнопка “Выйти” и раздел “Добавить рецепт”.
5. “Панель администратора” – страница где администратор может с помощью специальных форм удалять пользователей, выдавать и отнимать у них роль администратора, удалять рецепты.

## **5) Процесс работы по спринтам**

В ходе работы было запланировано и реализовано 4 спринта. Длительность спринта - одна неделя. За 4 спринта были выполнены основные поставленные задачи, в ходе работы и ретроспективы спринтов часть подзадач была перенесена на следующие спринты.

**Спринт I**

Цель: регистрация на хостинге и приобретение домена, создание репозитория, написание ТЗ. Цель достигнута.

Задачи:

* Выбрать тариф и приобрести услугу хостинга

Выбран и приобретён хостинг Oracle cloud, а также домен у провайдера Freenom.

* Создать репозиторий

На веб-сервере GitHub: https://github.com/YaroshenkoNick/-coursework

* Написать ТЗ

**Спринт II**

Цель спринта: создание целевой страницы сайта, разработать дизайн, создать базу данных. Цель достигнута.

Задачи:

* Разработать дизайн страниц сайта
* Написать php-код целевой страницы
* Прикрепить к header стилевой файл
* Составить разделы сайта
* Разработать и создать базу данных

**Спринт III**

Цель спринта: создание дополнительных страниц сайта с дизайном и формами регистрации и авторизации. Цель достигнута, возникли проблемы реализации форм регистрации и авторизации.

Задачи:

* Написать php-код дополнительных страниц
* Написать php-код формы регистрации
* Проанализировать общую картину сайта
* Подключить базу данных к сайту
* Разработать доп. страницы сайта

**Спринт IV**

Цель спринта: разделить пользовательские сценарии, создать форму добавления заданий для зарегистрированных пользователей, создать форму обратной связи, оформить страницы Категорий, где будут отображаться рецепты с подробным описанием, ингредиентами и фотографиями, разместить готовый сайт в сети Интернет. Сделать Цель спринта достигнута.

Задачи:

* Сделать вывод рецептов в страницы категорий.
* Разделить пользовательские сценарии в зависимости от того, зарегистрирован ли пользователь и какая у него роль
* Создать форму добавления рецептов
* Создать форму обратной связи
* Создать админ панель
* Зарегистрировать домен topfood.ddns.net
* Делегировать домен
* Выгрузить сайт на сервер

**Приложения**

Код формы добавления рецепта с изображением, его обработка для добавления в базу данных и вывод рецепта из базы данных на страницу

Код add\_recipe.php

<?php

session\_start();

$title = $\_POST['title'];

$category = $\_POST['category'];

$content = $\_POST['content'];

$ingredients = $\_POST['ingredients'];

require\_once 'create\_uid.php';

$result = $mysql->query("SELECT login FROM `users` WHERE `id` = '$uid'");

$take\_login = $result->fetch\_assoc();

$login = $take\_login['login'];

$res = $mysql->query("SELECT \* FROM `categories` WHERE `title` = '$category'");

$cnc = $res->fetch\_assoc();

if (empty($cnc)) {

$\_SESSION['message'] = 'Такой категории не существует' ;

header('Location: add\_recipe\_form.php');

} else {

$img\_type = substr($\_FILES['image\_recipe']['type'], 0, 5);

if (!empty($\_FILES['image\_recipe']['tmp\_name']) and $img\_type === 'image') {

global $img;

$img =

addslashes(file\_get\_contents($\_FILES['image\_recipe']['tmp\_name']));

}

$mysql->query("INSERT INTO `recipes` (`title`,`category\_title`,`content`,`image`,`ingredients`, `login`) VALUES('$title','$category','$content','$img', '$ingredients', '$login')");

$mysql->close();

$\_SESSION['message'] = 'Ваш рецепт был успешно добавлен' ;

header('Location: add\_recipe\_form.php'); }

?>

Код add\_recipe\_form.php

<?php

session\_start();

if (!$\_COOKIE['auth\_token']) {

header('Location: index.php');

}

require 'header.php';

?>

<div class="content" id="content\_add\_recipe">

<?php

if (isset($\_SESSION['message'])) {

echo '<div class="msg"> ' . $\_SESSION['message'] . ' </div>';

}

unset($\_SESSION['message']);

?>

<div class="main\_content" id="add\_recipe">

<div class="add\_recipe\_form">

<h2 id="h2\_add\_recipe">Добавьте ваш рецепт</h2>

<form action="add\_recipe.php" method="post" enctype="multipart/form-data">

<label for="title">Название:</label>

<input placeholder="Введите название" class="input\_add\_recipe" type="text" id="title" name="title" required>

<label for="category">Категория:</label>

<input placeholder="Введите категорию" class="input\_add\_recipe" type="text" id="category" name="category" required>

<label for="image\_recipe">Изображение:</label>

<input placeholder="Выберите изображение" class="input\_add\_recipe" type="file" id="image\_recipe" name="image\_recipe" required>

<textarea id="ingredients" name="ingredients" placeholder="Введите ингредиенты" maxlength="500" required></textarea>

<p id="p\_add\_recipe">Ингредиенты разделять переносом строки(Enter)</p>

<textarea id="describe\_add\_recipe" name="content" placeholder="Описание рецепта" maxlength="3000" required></textarea>

<button type="submit" class="input red\_button"><strong>Добавить</strong></button>

</form>

</div>

</div>

</div>

<?php require ('footer.php') ?>

Код first\_dish.php

<?php

session\_start();

require 'header.php';

?>

<div class="content">

<div class="main\_content" id="main\_content\_recipes\_display">

<h2>Закуски</h2>

<div class="recipes\_display">

<?php

require\_once 'db\_connection.php';

$result = $mysql->query("SELECT \* FROM `recipes` WHERE `category\_title`='Закуски' ORDER BY id DESC");

while ($row = $result->fetch\_assoc()) {

$show\_img = base64\_encode($row['image']); ?>

<div class="recipes\_display\_container">

<h2><?php echo $row['title']; ?></h2>

<div class="info\_recipe">

<div class="img\_ul\_recipes">

<img class="img\_recipes" src="data:image/jpeg;base64, <?php echo $show\_img ?>" alt="">

<ul>Ингридиенты: <?php $string = $row['ingredients'];

$tok = strtok($string, "\n");

while ($tok !== false) {

echo "<li>$tok</li>";

$tok = strtok("\n"); }

?>

</ul>

</div>

<div class="recipe\_content"><?php

$content = nl2br($row['content']);

echo $content;?></div>

<div class="author">Автор: <?php echo $row['login'];?></div>

</div>

<hr color="black" width="100%" size="3px"/>

<?php } ?>

</div>

</div>

</div>

</div>

<?php require 'footer.php' ?>

# **Вывод:**

В результате выполнения курсовой работой я познакомился с программой контроля версий Git, с веб-приложением phpmyadmin. Научился создавать базы данных, разрабатывать сайты, размещать сайты в сети интернет. Была разработана информационная система “ TopFood – сайт с рецептами блюд ”, которая разрешит проблему отсутствия полной и структурированной информации по предметам, которые изучаются в университете.