

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский горный университет

Кафедра Информационных систем и вычислительной техники
Производственная практика – технологическая (проектно-технологическая) – первая
производственная практика

ОТЧЕТ

Студента	Плотниковой Полины Сергеевны
Группа	ИАС-20
Личный код	200233
Направление подготовки	Информатика и вычислительная техника
Вид практики	производственная
Место проведения практики	Учебно-научный центр цифровых технологий Санкт-Петербургского горного университета, Санкт-Петербург, В. О. 21 линия д. 2
Сроки проведения практики	29.06.2023 – 26.07.2023
Должность практиканта	Программист
Руководитель от предприятия	Жуковский Ю.Л.
Руководитель от ВУЗа	Гурко А.В.

Студент

(подпись, дата)

(инициалы, фамилия)

Отметка руководителя практики от предприятия

В Отчете отсутствует информация конфиденциальная информация, связанная с предприятием.

Рекомендуемая оценка студенту

...

Руководитель практики от предприятия

(подпись, дата)

(инициалы, фамилия)

Отметка руководителя практики от университета

Итоговая Оценка

Руководитель практики от ВУЗа

(подпись, дата)

(инициалы, фамилия)

Содержание

ЗАДАНИЕ НА ПРАКТИКУ	3
ПАСПОРТ ПРОИЗВОДСТВЕННЫХ КОМПЕТЕНЦИЙ, СФОРМИРОВАННЫХ ПРИ ПРОХОЖДЕНИИ ПРАКТИЧЕСКОЙ ПОДГОТОВКИ В РАМКАХ.....	4
ВВЕДЕНИЕ	5
1. ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ.....	6
1.1 Общие сведения	6
1.2 Организация техники безопасности.....	7
2. ЗАДАЧИ, ВЫПОЛНЕННЫЕ ПРАКТИКАНТОМ СОГЛАСНО ДОЛЖНОСТИ	8
2.1 Общий обзор.....	8
2.2 Основной стек разработки.....	8
2.3 Создание рабочего окружения	10
2.4 Создание базы данных	10
2.5 Основной функционал web-игры.....	12
2.6 Демонстрация работы проекта.....	19
ДНЕВНИК.....	23
ВЫВОД.....	24
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	25

Задание на практику

**Паспорт производственных компетенций,
сформированных при прохождении практической подготовки в рамках
«Производственная практика - технологическая практика (проектно-технологическая
практика) - Вторая производственная практика»
по направлению подготовки 09.03.01 «Информатика и вычислительная техника»
профиль «Информационное и программное обеспечение автоматизированных систем»**

№п/п	Наименование производственных компетенций	Оценка уровня освоения компетенций, сформирована / не сформирована	Примечание
1.	Способность формировать алгоритмы разработки в соответствии с техническим заданием		
2.	Способность оценивать, выбирать и создавать варианты архитектуры проекта		
3.	Способность к самостоятельному обучению, умение развивать научно-производственный профиль своей профессиональной деятельности		
4.	Способность реализовывать и администрировать базы данных в конкретной системе управления базами данных		

Перечень производственных компетенций сформирован в полном объеме и соответствует программе «Производственной практики - технологической практики - Второй производственной практики»

Руководитель практики от предприятия

(подпись)

(ФИО)

Введение

Производственная практика представляет собой неотъемлемую часть учебного процесса, в ходе которого студенты имеют возможность применить и закрепить полученные теоретические знания на практике. В данном отчете будут представлены результаты и анализ практической работы, выполненной в рамках производственной практики. Отчет будет включать описание производственной среды, цели и задачи, поставленные перед студентом, а также методы и инструменты, использованные для достижения этих целей.

Целью практики является расширение представлений о профессиональной сфере деятельности, а также приобретение практических навыков и опыта. Поэтому основной целью практики стала разработка карточной web-игры «Мемогу».

Задачи практики:

1. Ознакомление с функциональными обязанностями предприятия.
2. Ознакомление с используемым на рабочем месте программным обеспечением, получение навыков работы.
3. Разработка web-игры «Мемогу», являющейся тренажером для запоминания минералов.
4. Приобретение необходимых компетенций.

1. Характеристика предприятия

1.1 Общие сведения

Центр «Цифровых технологий» - опорный научно-исследовательский и образовательный кластер для интеграции сквозных цифровых технологий и решений. Его сотрудники занимаются реализацией и исследованием наиболее актуальных и перспективных на сегодняшний день технологий и концепций. Среди них – цифровые двойники, большие данные, дополненная реальность, робототехника и численное моделирование.

Проекты, реализованные на базе программно-аппаратного комплекса, позволяют обеспечить экономический рост за счет цифровой трансформации в управлении объектами недропользования и энергетики.

Основной вычислительный узел Центра - сервер FORSITE 2U RS2-2049-24HS со следующими характеристиками: 112 ядер CPU, терабайт оперативной памяти, 10000 графических ядер CUDA. Кластер позволяет производить вычисления любой сложности с применением программного обеспечения ведущих компаний-разработчиков: Ansys, Rocky DEM, Flownex SE, Dassault Systemes, AVEVA Group.

В структуру Центра входят аудитории и лаборатории, созданные в сотрудничестве с такими компаниями как Новатэк, Caterpillar и Schneider Electric. Благодаря этим коллаборациям, студенты и партнеры вуза имеют возможность с помощью 3D- и VR-технологий обучаться и повышать компетенции в области увеличения эффективности электропотребления предприятий, управления добычей нефти и газа на шельфе, руководства горнотранспортными процессами и техникой.

Задачи центра:

1. Создание ключевых условий, способствующих подготовке высококвалифицированных кадров для цифровой экономики.
2. Внедрение цифровых технологий для модернизации компаний минерально-сырьевого и топливно-энергетического комплекса.
3. Обучение сотрудников компаний-партнеров работе с новыми продуктами цифровизации.
4. Развитие сотрудничества в области высоких технологий с профильными компаниями и другими университетами.
5. Пополнение научной базы в сфере цифровых технологий, публикация результатов исследований в научных журналах мирового масштаба.

1.2 Организация техники безопасности

1. Общие требования техники безопасности при работе на персональном компьютере

1.1. К работе на персональном компьютере допускаются лица, прошедшие обучение безопасным методам труда, вводный инструктаж, первичный инструктаж на рабочем месте.

1.2. При эксплуатации персонального компьютера на работника могут оказывать действие следующие опасные и вредные производственные факторы:

- повышенный уровень электромагнитных излучений;
- повышенный уровень статического электричества;
- пониженная ионизация воздуха;
- статические физические перегрузки;
- перенапряжение зрительных анализаторов.

1.3. Работник обязан:

1.3.1. Выполнять только ту работу, которая определена его должностной инструкцией.

1.3.2. Содержать в чистоте рабочее место.

1.3.3. Соблюдать режим труда и отдыха в зависимости от продолжительности, вида и категории трудовой деятельности.

1.3.3. Соблюдать меры пожарной безопасности.

1.4. Рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

1.5. Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, преимущественно слева.

2. требования техники безопасности в аварийных ситуациях

2.1. Во всех случаях обрыва проводов питания, неисправности заземления и других повреждений, появления гари, немедленно отключить питание и сообщить об аварийной ситуации руководителю.

2.2. Не приступать к работе до устранения неисправностей.

2.3. При получении травм или внезапном заболевании немедленно известить своего руководителя, организовать первую доврачебную помощь или вызвать скорую медицинскую помощь.

2. Задачи, выполненные практикантом согласно должности

Выполнены задачи:

1. Создание локального сервера для проекта;
2. Администрирование базы данных, подключение ее к проекту и использование;
3. Разработка алгоритмов, реализующих основную работу игры;
4. Создание пользовательского интерфейса игры.

2.1 Общий обзор

В рамках производственной практики была поставлена цель разработать карточную web-игру «Memory». Эта игра является классической игрой на память, где игроку предлагается переворачивать карточки на поле и искать парные изображения.

Игровое поле представляет собой сетку карточек, расположенных лицевой стороной вниз. Цель игрока состоит в том, чтобы открыть две карточки за один ход и найти парные изображения. Если изображения на карточках совпадают, они остаются открытыми, и игрок получает очки. Если изображения не совпадают, карточки переворачиваются обратно рубашкой вверх. Игра продолжается до тех пор, пока все парные карточки не будут найдены.

Количество очков, заработанных игроком, отображается на экране, что позволяет отслеживать свой прогресс. Кроме того, в игре реализованы анимации, которые делают игру более привлекательной и интерактивной. Игра также имеет функцию перезапуска, которая позволяет игроку начать игру заново.

Данная игра отличается от классической карточной игры на развитие памяти тем, что в качестве картинок используются изображения 32-х минералов. Каждому минералу присвоен балл по шкалы Мооса от 1 до 10, который также находится на карточке. Таким образом эта игра не только развивает память, но и может помочь студентам, которые учат дисциплины горнодобывающей промышленности, визуально запомнить минерал и его балл по шкале Мооса в развлекательной форме.

2.2 Основной стек разработки

Для web-игры использовались следующие инструменты:

- HTML (HyperText Markup Language) — это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

- JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений.
- CSS (Cascading Style Sheets) — формальный язык декорирования и описания внешнего вида документа (веб-страницы), используемый для визуального оформления веб-сайтов. Написан с использованием языка разметки (чаще всего HTML или XHTML).
- PHP (Hypertext Preprocessor) – C-подобный скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.
- MySQL – свободная реляционная система управления базами данных.
- PhpMyAdmin – веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд.
- Apache HTTP Server – свободный веб-сервер. Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, macOS, Microsoft Windows, Novell NetWare, BeOS. Он хорошо показывает себя в работе с масштабными проектами, поэтому заслуженно считается одним из самых популярных веб-серверов. Кроме того, Apache очень гибок в плане настройки, что даёт возможность реализовать все особенности размещаемого веб-ресурса.

2.3 Создание рабочего окружения

В рамках производственной практики был создан локальный сервер с помощью Apache, PHP, MySQL и phpMyAdmin. Локальный сервер – это набор программного обеспечения, который выполняет функции сервера, работающий на локальном компьютере или в локальной сети предприятия.

Такой подход может быть полезен, например, для разработчиков веб-сайтов, чтобы вначале создать общую структуру сайта, отладить его работу и отработать его дизайн у себя на компьютере, а затем перенести готовый к использованию веб-сайт на внешний веб-сервер, где он станет доступным для широкого круга пользователей.

Локальный сервер содержит, в принципе, то же программное обеспечение, что и обычный веб-сервер, только работает в пределах локальной сети или отдельного компьютера. Это означает, что в большинстве случаев он не будет подключен к Интернету, а если и будет, то будет защищен паролем, поэтому не каждый сможет получить доступ к его сервисам.

Созданный сервер предоставил окружение для разработки игры «Memory». Локальный сервер позволил эффективно работать над проектом, тестировать функциональность и вносить изменения при необходимости.

2.4 Создание базы данных

База данных в проекте «Memory» была разработана и использована для хранения различных данных, необходимых для функционирования игры. В базе данных была создана таблица «card», которая содержала информацию о минералах, такую как id минерала, его название, балл по шкале Мооса, изображение. Изображения минералов хранятся в проекте в папке minerals, а в базе данных записывается путь к каждому изображению. База данных обеспечивала целостность данных и облегчала работу с информацией, хранящейся в проекте.

Сервер: localhost » База данных: cards » Таблица: card									
Обзор Структура SQL Поиск Вставить Экспорт Импорт Прив									
Структура таблицы Связи									
#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	
<input type="checkbox"/>	1 id	int			Нет	Нен		AUTO_INCREMENT	
<input type="checkbox"/>	2 name	varchar(20)	utf8mb4_0900_ai_ci		Нет	Нен			
<input type="checkbox"/>	3 point	int			Нет	Нен			
<input type="checkbox"/>	4 img	varchar(100)	utf8mb4_0900_ai_ci		Нет	Нен			

Рисунок 1. Структура таблицы «card»

← T →				▼ id	name	point	img
<input type="checkbox"/>		Изменить		Копировать		Удалить	1 Тальк 1 \Practika 2023\minerals\1_1.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	2 Тальк 1 \Practika 2023\minerals\1_2.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	3 Тальк 1 \Practika 2023\minerals\1_3.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	4 Гипс 2 \Practika 2023\minerals\2_1.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	5 Гипс 2 \Practika 2023\minerals\2_2.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	6 Гипс 2 \Practika 2023\minerals\2_3.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	7 Кальцит 3 \Practika 2023\minerals\3_1.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	8 Кальцит 3 \Practika 2023\minerals\3_2.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	9 Кальцит 3 \Practika 2023\minerals\3_3.png
<input type="checkbox"/>		Изменить		Копировать		Удалить	10 Флюорит 4 \Practika 2023\minerals\4_1.png

Рисунок 2. Пример данных таблицы «card»

Для подключения базы данных к проекту был создан php-файл connect.php со следующим кодом:

```
<?php
$servername = "localhost";
$dbname = "cards";
$username = "root";
$password = "password";

// Создаем соединение
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Проверяем соединение
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Для получения нужной информации из базы данных был создан php-файл getdata.php со следующим кодом:

```
<?php
include 'connect.php';

// Выполнение запроса к базе данных
```

```

$sql = "SELECT * FROM `card`";
$result = $conn->query($sql);

// Получение данных из базы данных и формирование массива
$cards = array();
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $cards[] = array(
            'id' => $row['id'],
            'name' => $row['name'],
            'point' => $row['point'],
            'img' => $row['img']
        );
    }
}

// Закрытие соединения с базой данных
$conn->close();

// Возвращение данных в формате JSON
header('Content-Type: application/json; charset=utf-8');
echo json_encode($cards, JSON_UNESCAPED_UNICODE);
?>

```

Этот код возвращает данные в формате JSON, которые потом можно использовать в игре.

2.5 Основной функционал web-игры

Содержание файла index.html:

```

<!DOCTYPE html>

<html lang="ru" >

<head>

    <meta charset="UTF-8">

    <title>Найди пару</title>

```

```

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/meyer-
reset/2.0/reset.min.css">

        <link rel="stylesheet" href="./style.css">

</head>

<body>

    <div class="wrap">

        <div class="players">

            <h1>Игрок</h1>

            <div class="score">

                <h2>Score:</h2>

                <div class="new_point">0</div>

            </div>

        </div>

        <div>

            <!-- блок с игрой -->

            <div class="game"></div>

            <!-- модальное окно -->

            <div class="modal-overlay">

                <div class="modal">

                    <!-- поздравительная надпись -->

                    <h3 class="winner">Победа!</h3>

                    <button class="restart">Играть заново</button>

                </div>

            </div>

        </div>

    </div>

    <script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></script>

    <script src="./script.js"></script>

</body>

</html>

```

Содержание файла script.js и объяснение функционала игры:

// весь скрипт — это большая функция

(function(){

// объявляем объект, внутри которого будет происходить основная механика игры

var Memory = {

// создаём карточку

init: function(cards){

// получаем доступ к классам

this.\$wrap = \$(".wrap");

this.\$game = \$(".game");

this.\$modal = \$(".modal");

this.\$overlay = \$(".modal-overlay");

this.\$restartButton = \$(".button.restart");

//собираем из карточек массив — игровое поле. merge - слияние двух массивов в один

this.cardsArray = \$.merge(cards, cards);

// перемешиваем карточки

this.shuffleCards(this.cardsArray);

// и раскладываем их

this.setup();

},

// метод перемешивания карточек

shuffleCards: function(cardsArray){

// используем встроенный метод .shuffle

this.\$cards = \$(this.shuffle(this.cardsArray));

},

// тасование Фишера–Йетса

shuffle: function(array){

var counter = array.length, temp, index;

while (counter > 0) {

index = Math.floor(Math.random() * counter);

```

counter--;
temp = array[counter];
array[counter] = array[index];
array[index] = temp;
    }
    return array;
},

```

// раскладываем карты

```

setup: function(){
    // код с карточками
    this.html = this.buildHTML();
    // добавляем код в блок с игрой
    this.$game.html(this.html);
    // доступ к сформированным карточкам
    this.$memoryCards = $(".card");
    // переворот второй карточки
    this.card_2 = false;
    // переворот первой карточки
    this.card_1 = null;
    // балл карты
    this.now_point = null;
// добавляем элементам на странице реакции на нажатия
    this.binding();
},
// реакция на нажатия
binding: function(){
    // нажатие на карточку
    this.$memoryCards.on("click", this.cardClicked);
    // нажатие на кнопку перезапуска игры
    this.$restartButton.on("click", $.proxy(this.reset, this));
},

```

```

cardClicked: function(){
    var _ = Memory;
    var $card = $(this);

    // если карточка не перевёрнута и мы не нажимаем на ту же
самую карточку второй раз подряд
    if(!_card_2 && !$card.find(".inside").hasClass("matched") &&
!$card.find(".inside").hasClass("picked")){

        // переворачиваем её
        $card.find(".inside").addClass("picked");
        if(!_card_1){
            // сохраняем первую карточку
            _card_1 = $(this).attr("data-id");

            // если мы перевернули вторую и она совпадает с
первой
        } else if(_card_1 == $(this).attr("data-id") &&
!$(this).hasClass("picked")){

            // оставляем обе на поле перевёрнутыми и
показываем анимацию совпадения
            $(".picked").addClass("matched");

            //сохраняем балл карточки
            _now_point=$(this).find(".card-point").attr("data-
pt");

            //добавляем балл игроку и отображаем его
            _score(_now_point);

            // обнуляем первую карточку
            _card_1 = null;

            // если вторая не совпадает с первой
        } else {

            // обнуляем первую карточку
            _card_1 = null;

            //не ждём переворота второй
карточки
            _card_2 = true;

            //ждём полсекунды и
переворачиваем всё обратно

```



```

        setTimeout(function(){
            $(".picked").removeClass("picked");
            Memory.card_2 = false;
        }, 600);
    }

    // если перевернули все карточки
    if($(".matched").length == $(".card").length){
        // показываем победное сообщение
        _.win();
    }},

//добавляем баллы игроку и отображаем их
score: function(newpoint){
    var newPointElement = document.querySelector('.new_point');
    var currentNumber = parseInt(newPointElement.textContent);
    var currentPoint = parseInt(newpoint);
    var newNumber = currentNumber + currentPoint;
    newPointElement.textContent = Number(newNumber);
},

//обнуляем баллы игрока
null_score: function(){
    var newPointElement = document.querySelector('.new_point');
    newPointElement.textContent = 0;
},

// показываем победное сообщение
win: function(){
    this.card_2 = true;

    // плавно показываем модальное окно
    setTimeout(function(){
        Memory.showModal();
        Memory.$game.fadeOut();
    }, 1000);
},

```

```

// показываем модальное окно
showModal: function(){
    // плавно делаем блок с сообщением видимым
    this.$overlay.show();
    this.$modal.fadeIn("slow");
},
// убираем модальное окно
hideModal: function(){
    this.$overlay.hide();
    this.$modal.hide();
},
// перезапускаем игру
reset: function(){
    //обнуляем баллы игрока
    this.null_score();
    // прячем модальное окно с поздравлением
    this.hideModal();
    // перемешиваем карточки
    this.shuffleCards(this.cardsArray);
    // раскладываем карточки
    this.setup();
    // показываем игровое поле
    this.$game.show("slow");
},
// добавление карточек на страницу
buildHTML: function(){
    var frag = "";
    this.$cards.each(function(k, v){
        frag += '<div class="card" data-id="'+ v.id +'">\
            <div class="inside">\
                <div class="front">\

```

```

        <div class="card-point" data-pt="'+
            v.point +'>\
        '+ v.point +'\'
    </div>\
    \'
    </div>\
    <div class="back"></div>\
</div>\
</div>';
    });
    return frag;
    });

var cards = [];

// Получение данных из базы данных через AJAX-запрос
var xhr = new XMLHttpRequest();
xhr.open('GET', 'getdata.php', true);
xhr.onreadystatechange = function() {
    if (xhr.readyState === 4 && xhr.status === 200) {
        cards = JSON.parse(xhr.responseText);
        // запускаем игру
        Memory.init(cards);
    }
};
xhr.send();
})();

```

2.6 Демонстрация работы проекта

При запуске игры на экране появляются 2 блока – блок игрока со счетом баллов и блок с карточками. Карточек на поле 64 штуки, располагаются плиткой 8x8.

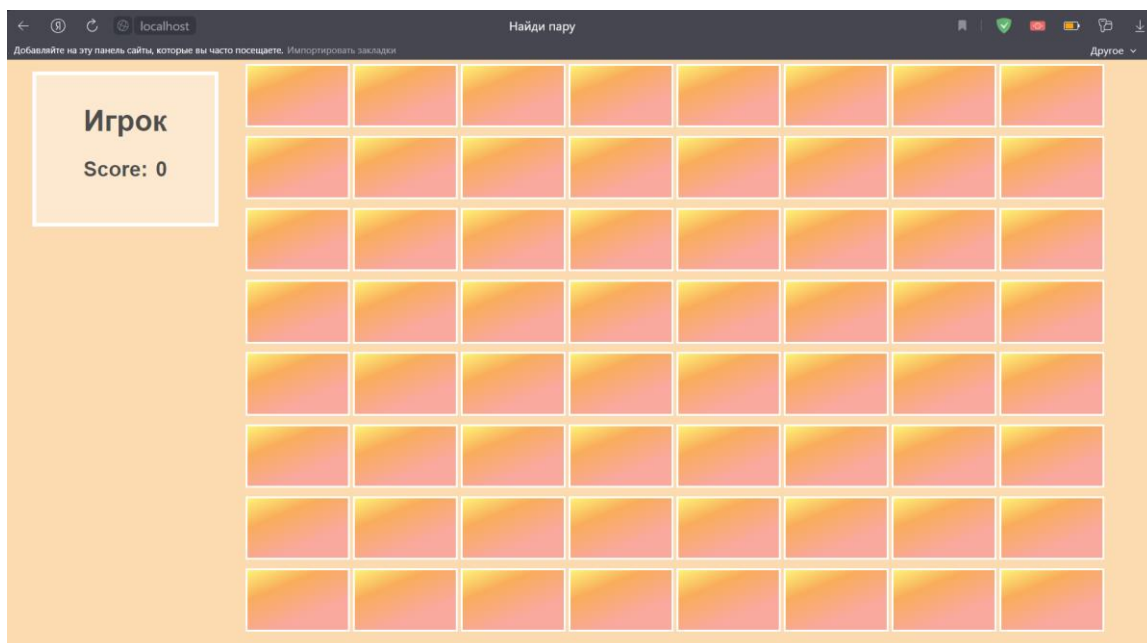


Рисунок 3. Игровое поле

При нажатии на карточку, она переворачивается внутренней стороной с изображением минерала и балла по шкале Мооса. Если при нажатии на вторую не открытую карточку оказывается, что на ней другой минерал, то обе переворачиваются обратно. Если минералы совпадают, то карточки на мгновение загораются зеленым цветом и остаются на поле открытыми, а игроку начисляется балл, написанный на карточке.

Для реализации переворачивания карточек в игре "Memory" было использовано свойство CSS-стилей "transform" с анимацией. При клике на карточку, ей применяется эффект переворота с помощью CSS-преобразования. Это создает иллюзию, что карточка переворачивается, и игрок может видеть изображение, которое находится на обратной стороне карточки.

Кроме того, для организации расположения карточек на игровом поле был использован CSS-модуль "flex". Flexbox позволяет легко управлять расположением элементов на странице и создавать гибкую и адаптивную структуру. Использование flex позволило создать гибкую и адаптивную структуру игрового поля, где карточки могут автоматически менять свое расположение в зависимости от размера экрана или изменений размеров окна браузера. Это обеспечивает хорошую пользовательскую экспериментирование и улучшает игровой процесс.

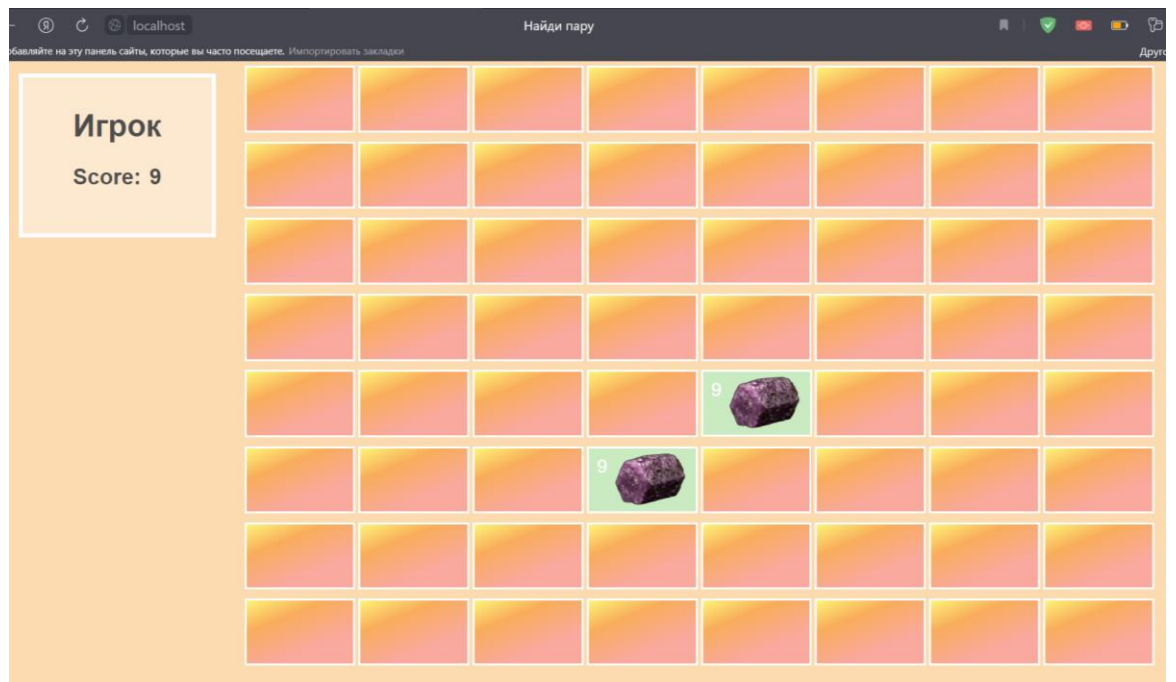


Рисунок 4. Нахождение пары

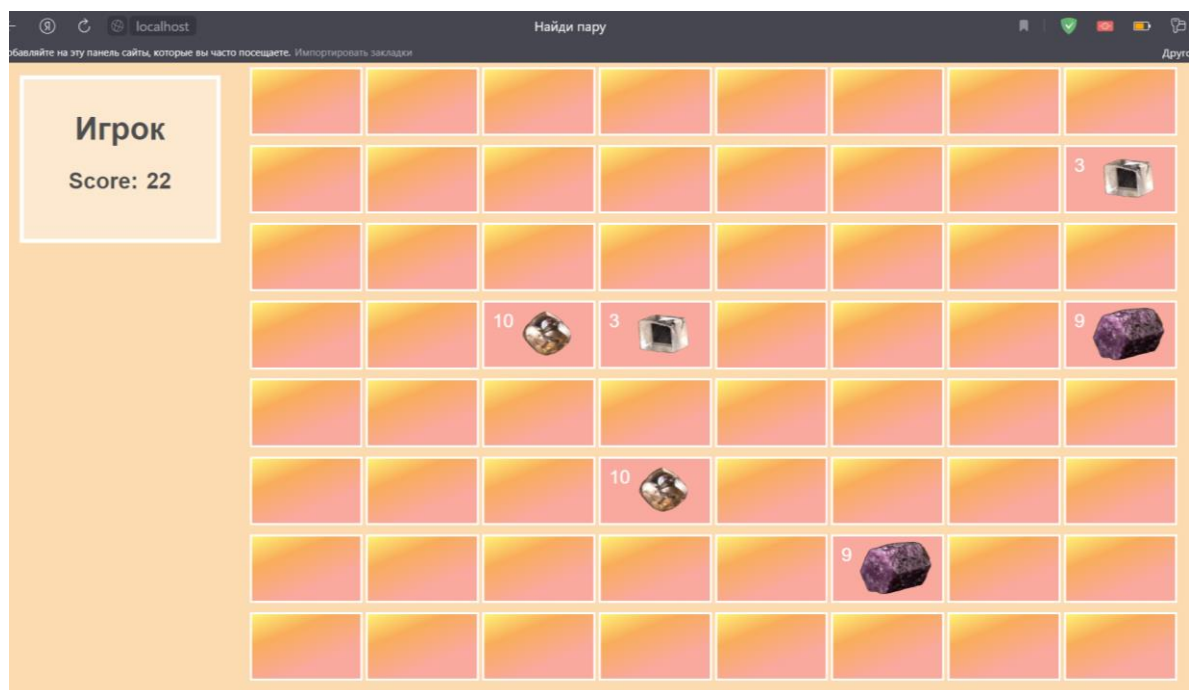


Рисунок 4. Нахождение нескольких пар и подсчет баллов

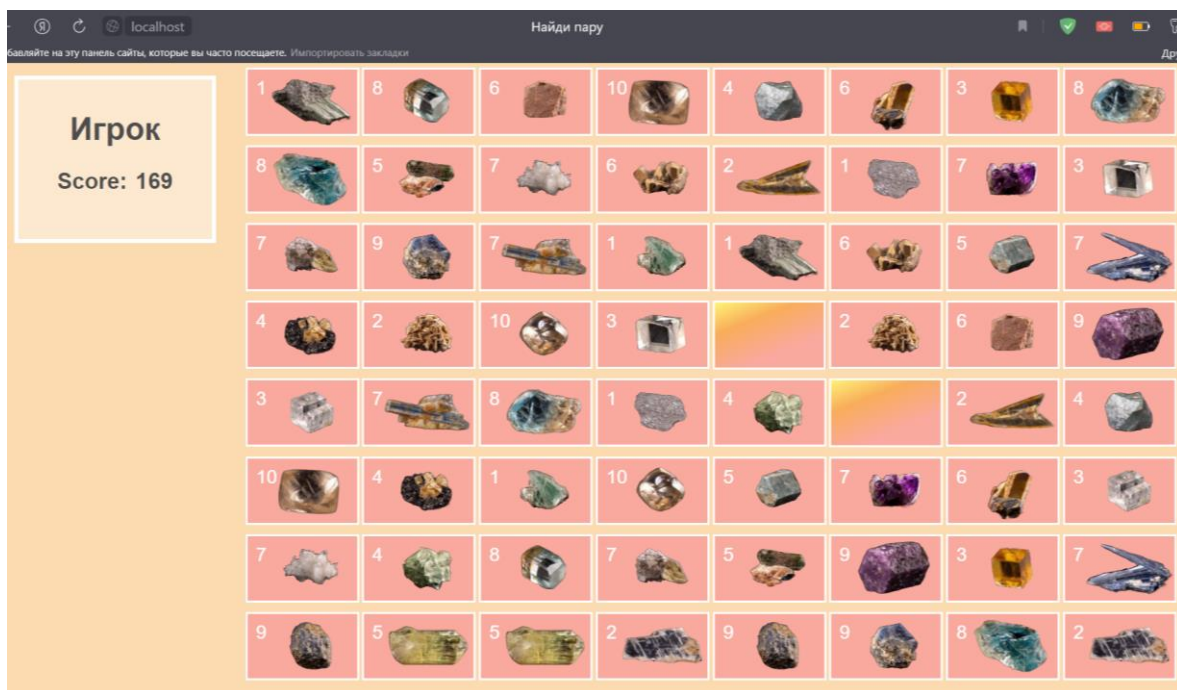


Рисунок 5. Вид почти оконченной игры

После того, как будут найдены все пары минералов, игра закончится и отобразится окно о победе и предложении сыграть заново. При нажатии на кнопку «Играть заново» игра перезапустится: карточки снова в разном порядке лягут «рубашкой» вверх, а баллы игрока обнулятся.

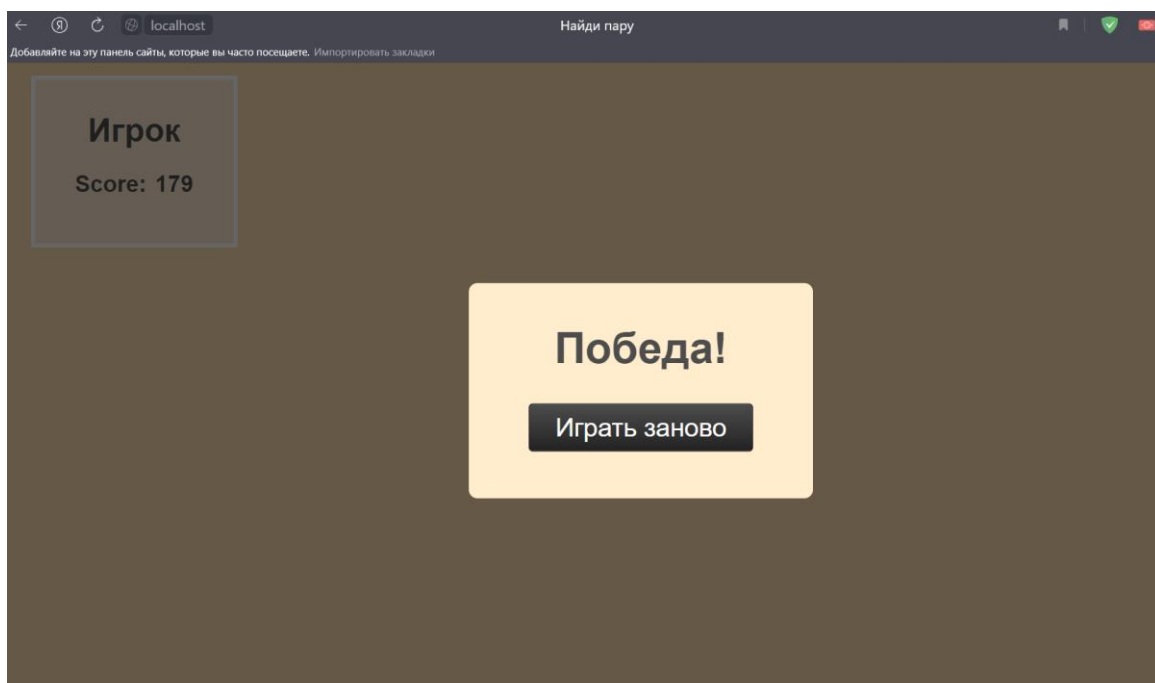


Рисунок 6. Объявление о победе

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский горный университет

Кафедра Информационных систем и вычислительной техники
Производственная практика – технологическая (проектно-технологическая) – вторая производственная практика

ДНЕВНИК

Студента	Плотниковой Полины Сергеевны
Группа	ИАС-20
Личный код	200233
Направление подготовки	Информатика и вычислительная техника
Вид практики	производственная
Место проведения практики	Учебно-научный центр цифровых технологий Санкт-Петербургского горного университета, Санкт-Петербург, В. О. 21 линия д. 2
Сроки проведения практики	29.06.2023 – 26.07.2023
Должность практиканта	Программист
Руководитель от предприятия	Жуковский Ю.Л.
Руководитель от ВУЗа	Гурко А.В.

ВЫПОЛНЕННАЯ РАБОТА

Дата	Содержание выполненной работы	Отметка о выполнении
29.06	Прохождение основных инструктажей на рабочем месте. Ознакомление с организационной структурой предприятия и ключевыми видами деятельности организации.	
30.06	Планирование архитектуры проекта, подбор стека разработки	
3.07 – 4.07	Создание рабочего окружения	
5.07	Создание базы данных	
6.07 – 7.07	Подключение базы данных к проекту и вывод значений	
10.07 – 14.07, 17.07 – 20.07	Разработка алгоритмов функционирования игры на JavaScript	
21.07, 24.07 – 25.07	Создание пользовательского интерфейса	
26.07	Написание отчёта о прохождении практики.	

Студент

(подпись, дата)

(инициалы, фамилия)

Вывод

В ходе практики были приобретены навыки в работе с веб-технологиями, базами данных и настройкой сервера. Были улучшены навыки программирования, а также получилось овладеть новыми концепциями и методиками разработки веб-приложений.

Игра "Memory" и настройка локального сервера с использованием Apache, PHP, MySQL и phpMyAdmin были успешно выполнены. Разработка проекта позволила практически применить свои знания и навыки, а также получить ценный опыт работы с веб-технологиями, базами данных и в создании игровых приложений.

Кроме вышеперечисленного, ежедневно вносились записи о проделанной работе в дневник практики.

Считаю необходимым и дальше расширять свои знания в области web-программирования, в частности глубже изучить работу с языком программирования JavaScript и серверными технологиями.

В ходе прохождения практики были выполнены следующие задачи:

1. Создание локального сервера;
2. Создание и работа с базой данных;
3. Разработка алгоритмов игры и представление ее в пользовательском интерфейсе.

Список использованной литературы

1. Гарретт Д. Веб-дизайн. Элементы опыта взаимодействия / Д. Гарретт. — СПб.: Символ-плюс, 2015
2. 2. Бер, Бибо jQuery. Подробное руководство по продвинутому JavaScript / Бибо Бер. - М.: Символ-плюс, 2015
3. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. - М.: Питер, 2015
4. Хаббард, Дж. Автоматизированное проектирование баз данных / Дж. Хаббард. - М.: Мир, 2013
5. Колисниченко Д.Н.: Современный сайт на PHP и JavaScript. - СПб.: Питер, 2009