Министерство образования и науки Республики Казахстан

Северо-Казахстанский университет им. М. Козыбаева

СЕМЕСТРОВЫЙ ПРОЕКТ

На тему: «3D порталы»

ИКТ1ИС.01 СП

|  |  |
| --- | --- |
| Выполнили студенты  группы АПО-19 | Хандрико К.В.  Сахаров А.В.  Буряк Р.С. |
| Научный руководитель  доцент, PhD | Астапенко Н.В. |

Петропавловск, 2022

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc102119928)

[1. Выбор методологии. 4](#_Toc102119929)

[2. Распределение обязанностей 5](#_Toc102119930)

[3. Разработка базы данных. 6](#_Toc102119931)

[4. Архитектура проекта 7](#_Toc102119932)

[5. Разработка алгоритма и интерфейса основных страниц 7](#_Toc102119933)

[6. Разработка порталов 11](#_Toc102119934)

[7. Разработка функций регистрации и авторизации 12](#_Toc102119935)

[ЗАКЛЮЧЕНИЕ 13](#_Toc102119936)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 14](#_Toc102119937)

[Код программы 15](#_Toc102119938)

## ВВЕДЕНИЕ

Актуальность: A-Frame - это веб-фреймворк с открытым исходным кодом для создания виртуальной реальности. Он поддерживается разработчиками из Supermedium и Google. A-Frame - это системная структура сущностных компонентов для Three.js, где разработчики могут создавать сцены 3D и WebVR, используя HTML

A-Frame показал себя не с самой лучшей стороны, когда стандарт HTML предлагал вариант frame. Ситуация ненамного стала лучше с появлением iframe. Браузеры и владельцы сайтов не слишком охотно соглашаются на отображение сторонней информации посредством таких элементов.

A-Frame это окно в мир данных, которые кто-то собирает, обрабатывает и представляет на своем веб-ресурсе. Также предлагает к ним доступ через фрейм на сайте-потребителе, как правило, через систему требований или специализированное API (библиотеку функций).

Далеко не каждый сайт может быть отображен в iframe. Прежде чем планировать на своем сайте использование фрейма, разработчик должен согласовать это с владельцем веб-ресурса или документа, которые он собирается отобразить на своем сайте в том или ином виде.

Темой данного проекта является разработка проекта «3D порталы». Для реализации данной задачи была выбран iFrame – встроенный фрейм в стандарте HTML

Проект «Square» создан для предоставления информации в интересной, интерактивной форме.

Цель проекта: продемонстрировать работу 3D порталов в Aframe.

Для реализации поставленной цели необходимо решить следующие задачи:

* выбрать методологию разработки;
* распределить задачи;
* разработать базу данных;
* разработать алгоритм;
* разработать интерфейс основных страниц;
* разработать порталы;
* разработать функции регистрации и авторизации.

Практическая значимость: Реализация данного проекта наглядно демонстрирует возможности Aframe разработки, в частности реализацию 3D порталов, главное преимущество которых это демонстрация предпросмотра страницы.

## **Выбор методологии.**

Метод разработки динамических систем (Dynamic Systems Development Method, DSDM) — это главным образом методика разработки программного обеспечения, основанная на концепции быстрой разработки приложений (Rapid Application Development, RAD). В 2007 году DSDM стал основным подходом к управлению проектом и разработки приложений. DSDM — это итеративный и инкрементный подход, который придаёт особое значение продолжительному участию в процессе пользователя/потребителя.

Цель метода — сдать готовый проект вовремя и уложиться в бюджет, но в то же время регулируя изменения требований к проекту во время его разработки. DSDM входит в семейство гибкой методологии разработки программного обеспечения, а также разработок не входящих в сферу информационных технологий.

Существует 9 принципов, состоящих из 4 основных и 5 начальных точек.

Вовлечение пользователя — это основа ведения эффективного проекта, где разработчики делят с пользователями рабочее пространство и поэтому принимаемые решения будут более точными.

Команда должна быть уполномочена принимать важные для проекта решения без согласования с начальством.

Частая поставка версий результата, с учётом такого правила, что «поставить что-то хорошее раньше — это всегда лучше, чем поставить всё идеально сделанное в конце». Анализ поставок версий с предыдущей итерации учитывается на последующей.

Главный критерий — как можно более быстрая поставка программного обеспечения, которое удовлетворяет текущим потребностям рынка. Но в то же время поставка продукта, который удовлетворяет потребностям рынка, менее важна, чем решение критических проблем в функционале продукта.

Разработка — итеративная и инкрементная. Она основывается на обратной связи с пользователем, чтобы достичь оптимального с экономической точки зрения решения.

Любые изменения во время разработки — обратимы.

Требования устанавливаются на высоком уровне прежде, чем начнётся проект.

Тестирование интегрировано в жизненный цикл разработки.

Взаимодействие и сотрудничество между всеми участниками необходимо для его эффективности.

Выбор этой методологии поможет в успешной разработке проекта.

## 2. Распределение обязанностей

Во втором этапе работы была поставленная следующая задача – распределение обязанностей между участниками проекта.

|  |  |
| --- | --- |
| Разработка проекта | |
| Студент | Обязанности |
| Хандрико К.В. | Создание базы данных, разработка порталов, добавление 3D моделей. |
| Сахаров А.В. | Создание и оформление 3D страниц, добавление аудио сопровождения на страницах, |
| Буряк Р.С. | Кодинг php запросов к базе данных, создание страниц авторизации и регистрации, добавление текста по клику на модели. |

|  |  |
| --- | --- |
| Выполнение отчета | |
| Студент | Обязанности |
| Хандрико К.В. | Описание выбранной методологии, распределение обязанностей, описание 3D моделей и порталов |
| Сахаров А.В. | Введение, описание оформления 3D страниц. |
| Буряк Р.С. | Заключение, архитектура проекта и список использованной литературы, описание функций появления текста. |

Таблица 1. Распределение обязанностей

## Разработка базы данных.

Проект подразумевает использование базы данных в качестве хранения пользовательских данных, таких как: логин, пароль, hash, ip, выбор страницы.

В качестве базы данных был выбран mysql, веб-приложение phpmyadmin.

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.

Приложение распространяется под лицензией GNU General Public License и поэтому многие другие разработчики интегрируют его в свои разработки, например XAMPP, Denwer, AppServ, Open Server.

Для реализация функции регистрации и авторизации была создана база данных testtable, в которой создана таблица users. Таблица users содержит 6 колонок:

user\_id - содержит уникальный идентификатор пользователя

user\_login – уникальный логин пользователя

user\_password – уникальный пароль пользователя

user\_hash – случайно образованный хэш

user\_ip – ip пользователя

choise – выбранная пользователем страница

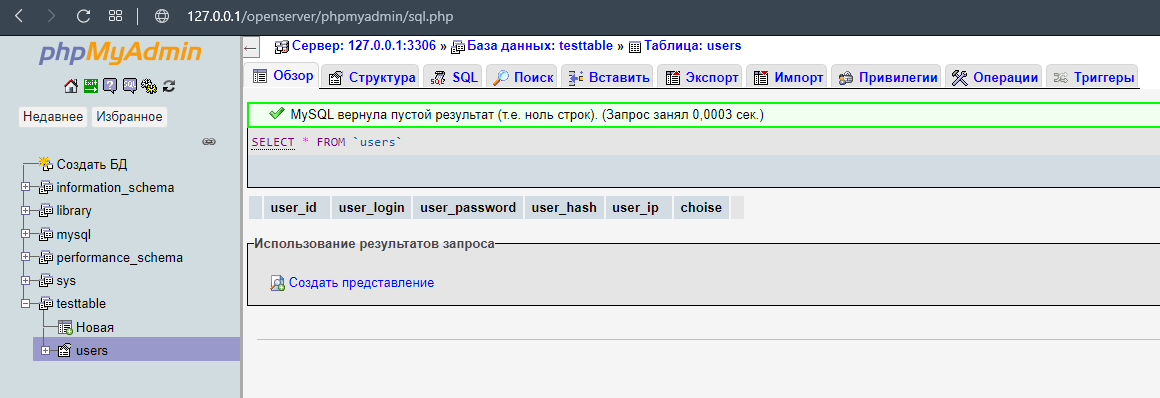
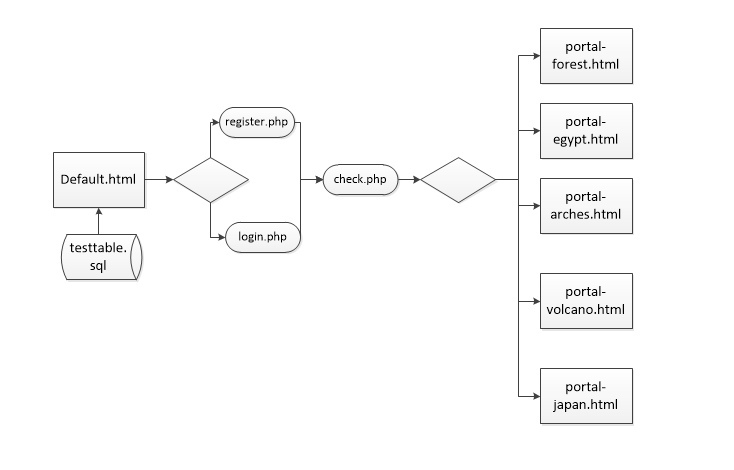


Рисунок 1. База данных testtable

## Архитектура проекта



В ходе выполнения проекта была разработана данная архитектура.

## Разработка алгоритма и интерфейса основных страниц

Как было написано ранее, задумкой проекта является несколько различных страниц которые содержут рельеф некой местности, а также страницы регистрации и авторизации.

Мы начали с разработки страницы на которой будут расположены порталы авторизации и регистрации.

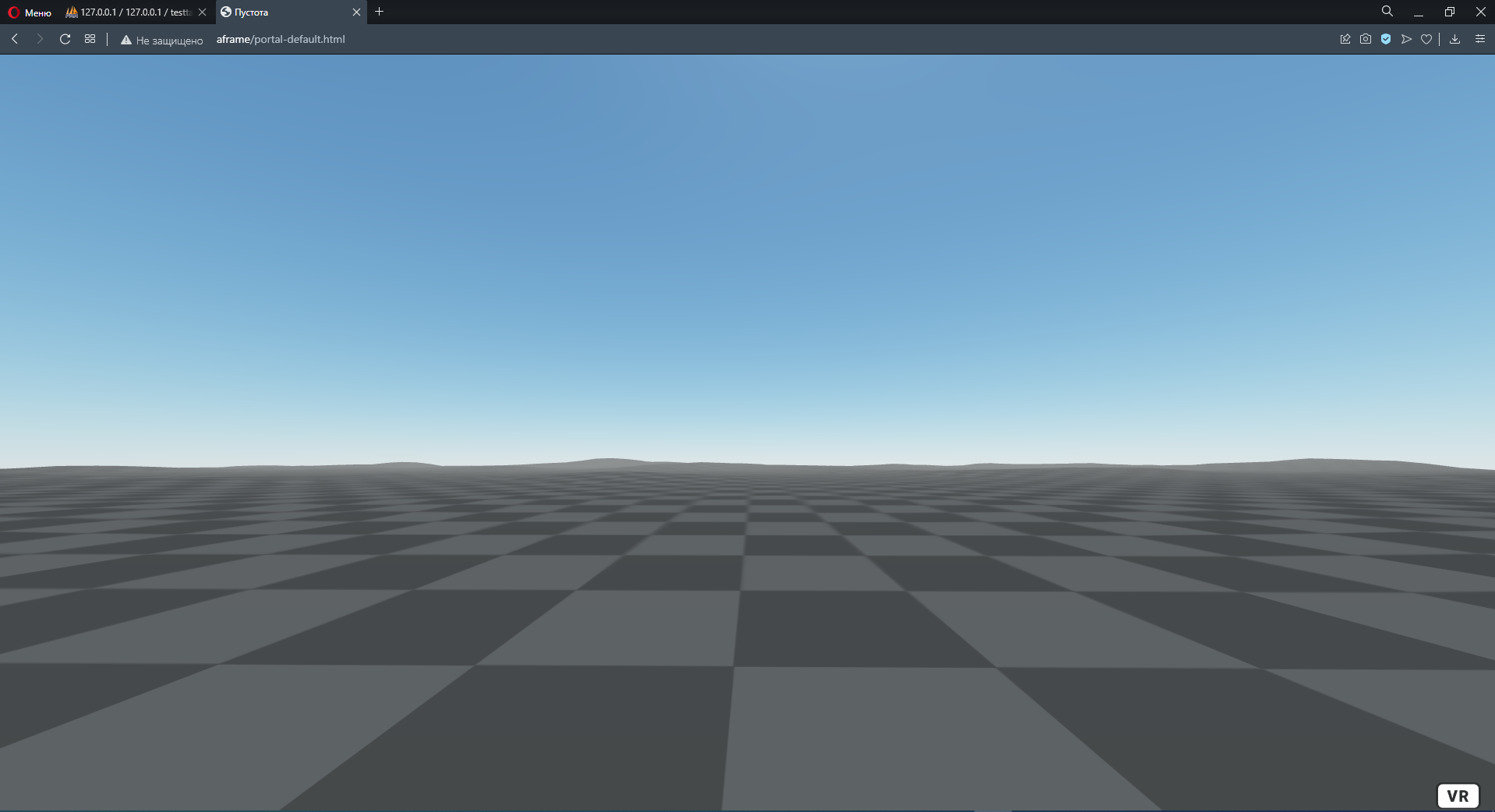


Рисунок 2. Начальная страница

Следующим этапом была разработка страниц с различным рельефом.

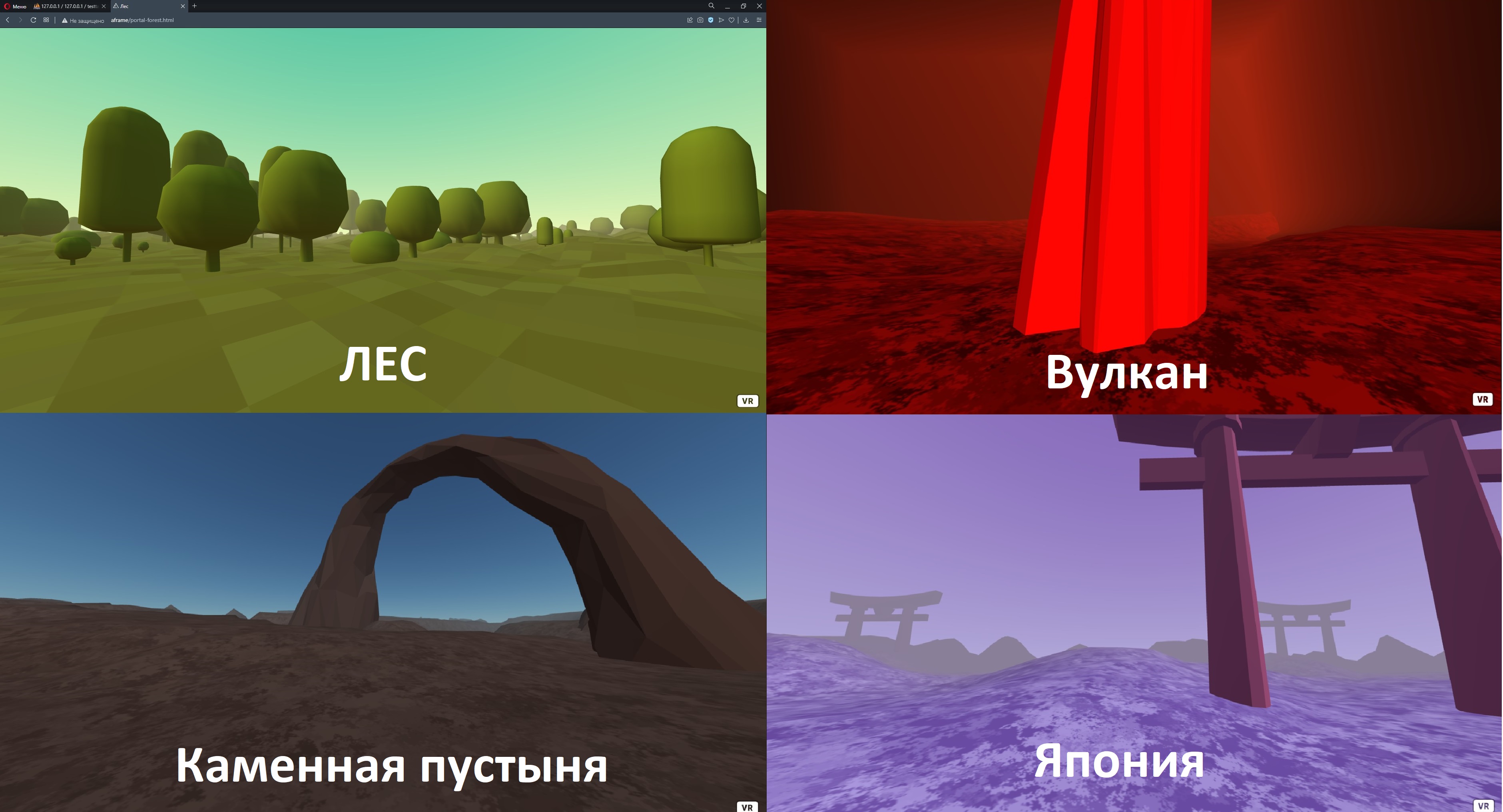


Рисунок 3. Основные страницы.

Страницы регистрации и авторизации имеют одинаковый вид

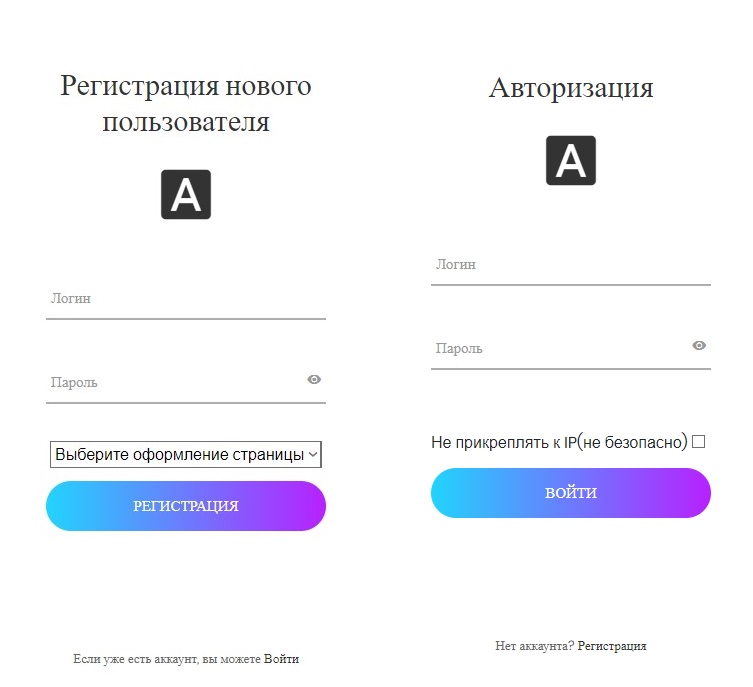


Рисунок 4. Страницы регистрации и авторизации.

После разработки непосредственно интерфейса страниц добавлены на каждую страницу 3D модельки.

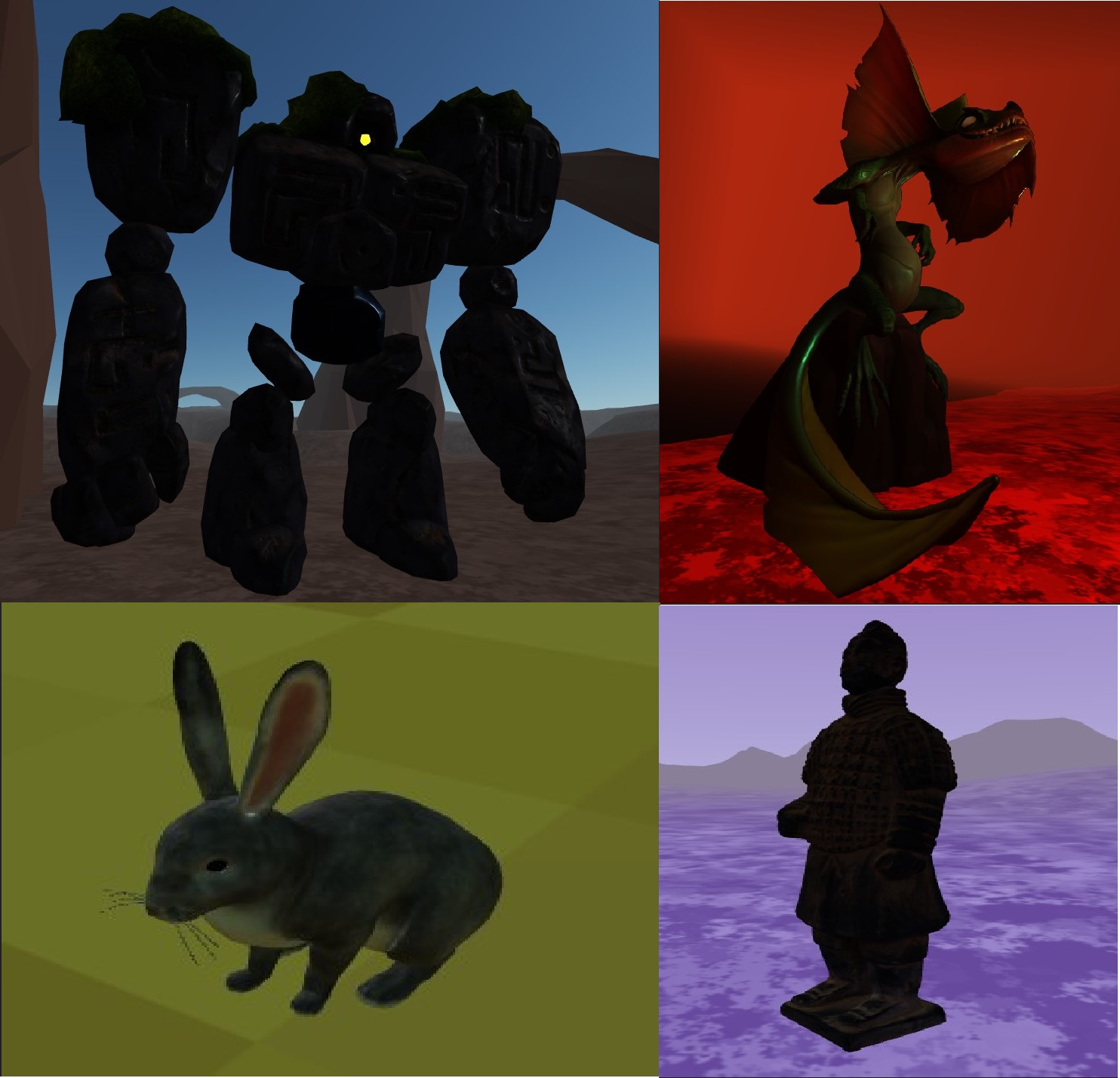


Рисунок 5. 3D модели

При нажатии на 3D модель появляется соответствующая надпись.



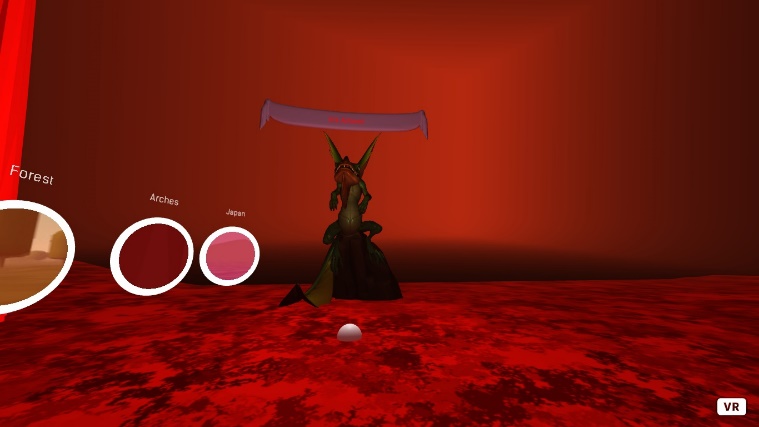
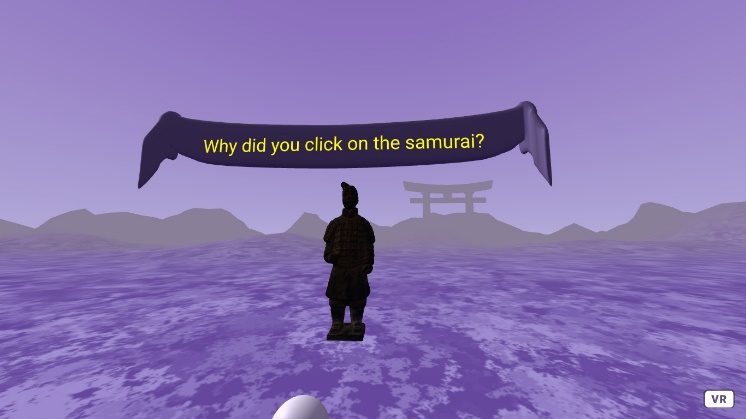


Рисунок 6. Надписи после нажатия на кнопки

На данных рисунках изображено появление текста после нажатия на соответствующие 3D модели.

## Разработка порталов

Нашей главной задумкой проекта были порталы, по сути они являются просто гиперссылкой, кликая по которой мы переходим на другую страницу. Но в Aframe их главной особенностью является то что эти порталы объемны и мы можем поместить в них любую правильно отмасштабированную картинку. Это будет выглядеть очень эффектно и впечатляюще. Мы разрабатываем такие порталы на каждой странице нашего проекта.

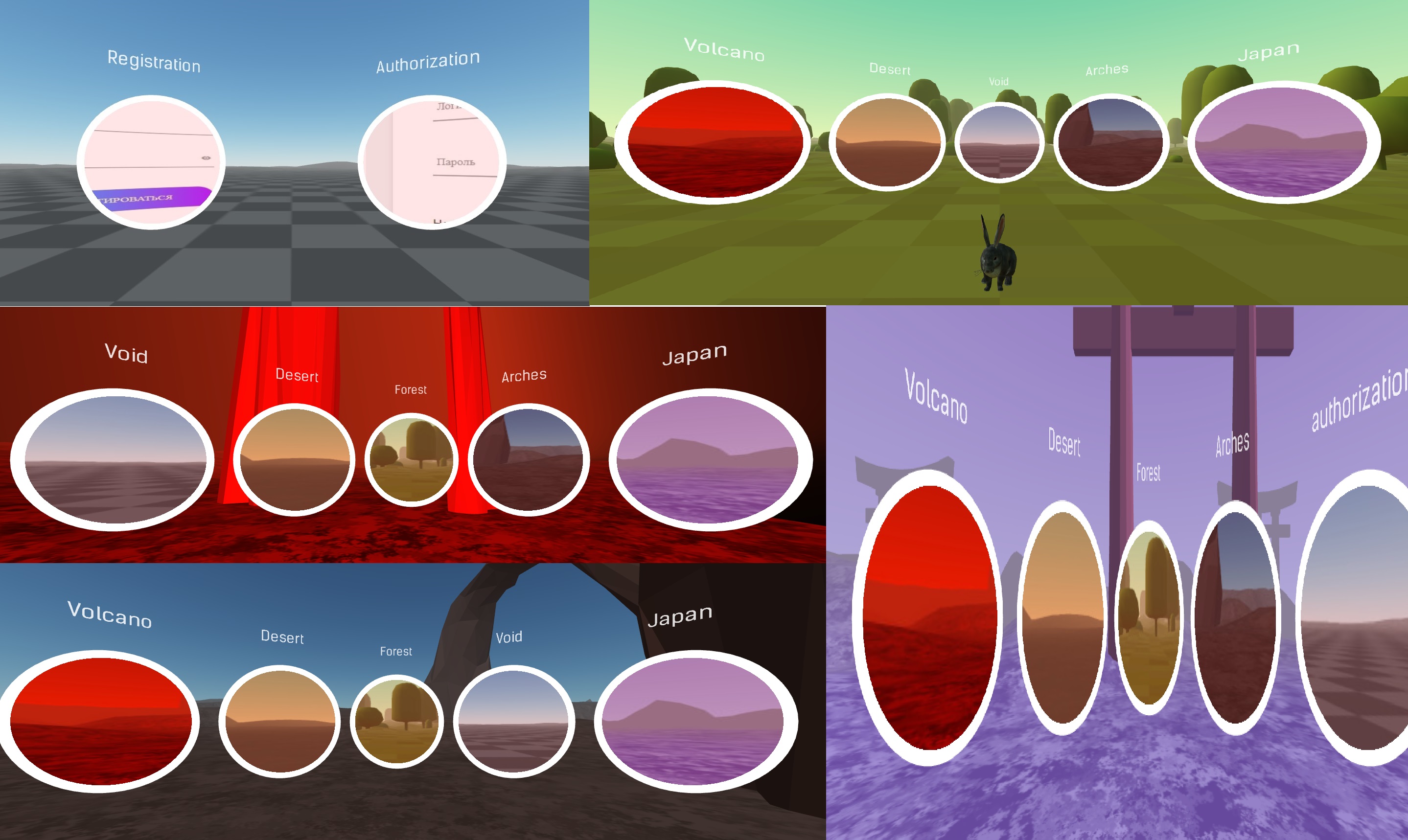


Рисунок 7. Порталы

## Разработка функций регистрации и авторизации

Модель авторизации:

Клиент

Регистрация:

* логин (a-z0-9)
* пароль

Вход:

* логин
* пароль

Cookie:

* уникальный идентификатор юзера
* хэш

Таблица users  
user\_id (int(11))  
user\_login (Varchar(30))  
user\_password (varchar(32))  
user\_hash (varchar(32))  
user\_ip (int(10)) по умолчанию 0

При регистрации в базу данных записывается логин пользователя и пароль(в двойном md5 шифровании)

При авторизация, сравнивается логин и пароль, если они верны, то генерируется случайная строка, которая хешируется и добавляется в БД в строку user\_hash. Также записывается IP адрес пользователя(но это у нас будет опциональным, так как кто-то сидит через Proxy, а у кого-то IP динамический... тут уже пользователь сам будет выбирать безопасность или удобство). В куки пользователя мы записываем его уникальный идентификатор и сгенерированный hash.

Почему надо хранить в куках хеш случайно сгенерированной строки, а не хеш пароля?  
1. Из-за невнимательности программиста, во всей системе могут быть дырки, воспользовавшись этими дырками, злоумышленник может вытащить хеш пароля из БД и подставить его в свои куки, тем самым получить доступ к закрытым данным. В нашем же случае, двойной хеш пароля не чем не сможет помочь хакеру, так как расшифровать он его не сможет(теоретически это возможно, но на это он потратит не один месяц, а может быть и год) а воспользоваться этим хешем ему негде, ведь у нас при авторизации свой уникальный хеш прикрепленный к IP пользователя.  
2. Если злоумышленник вытащит трояном у пользователя уникальный хеш, воспользоваться им он также не сможет(разве если только, пользователь решил пренебречь своей безопасностью и выключил привязку к IP при авторизации).

## ЗАКЛЮЧЕНИЕ

Выполнение проекта «3D порталы» наглядно продемонстрировало такие интересную функцую Aframe как порталы, будущем при выполнение похожих проектов, это позволит использовать порталы в более объемных информационных 3D полях. По проекту были поставлены задачи:

* выбрать методологию разработки;
* распределить задачи;
* разработать алгоритм;
* разработать базу данных;
* разработать интерфейс основных страниц;
* разработать порталы;
* разработать функции регистрации и авторизации.

В ходе выполнения проектной работы все задачи были выполнены в полном объеме.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. [Электронный ресурс]. – Режим доступа: URL: https://www.youtube.com/watch?v=ktjMCanKNLk&list=PL8MkBHej75fJD-HveDzm4xKrciC5VfYuV&ab\_channel=DaniloPasquariello. (23.02.2022)
2. [Электронный ресурс]. – Режим доступа: URL: https://www.youtube.com/watch?v=hyUBMmL0WtA. (23.02.2022)
3. [Электронный ресурс]. – Режим доступа: URL: https://www.youtube.com/watch?v=fYFiQ7lpfiE. (23.02.2022)
4. [Электронный ресурс]. – Режим доступа: URL: https://www.youtube.com/watch?v=JfpCicDUMKc. (23.02.2022)
5. [Электронный ресурс]. – Режим доступа: URL: https://www.youtube.com/watch?v=1xkWYCJaBAU. (23.02.2022)
6. [Электронный ресурс]. – Режим доступа: URL: https://www.youtube.com/watch?v=NijFSs03Pd4. (23.02.2022)
7. [Электронный ресурс]. – Режим доступа: URL: https://stepik.org/lesson/349987/step/1?unit=333841. (23.02.2022)
8. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»)

## Код программы

Структура таблицы `users` в базе данных 'testtable'

CREATE TABLE `users` (

    `user\_id` int(11) unsigned NOT NULL auto\_increment,

    `user\_login` varchar(30) NOT NULL,

    `user\_password` varchar(32) NOT NULL,

    `user\_hash` varchar(32) NOT NULL default '',

    `user\_ip` int(10) unsigned NOT NULL default '0',

    PRIMARY KEY (`user\_id`)

) ENGINE=MyISAM DEFAULT CHARSET=cp1251 AUTO\_INCREMENT=1 ;

register.php

<?

// Страница регистрации нового пользователя

// Соединямся с БД

$link=mysqli\_connect("localhost", "mysql\_user", "mysql\_password", "testtable");

if(isset($\_POST['submit']))

{

$err = [];

// проверям логин

if(!preg\_match("/^[a-zA-Z0-9]+$/",$\_POST['login']))

{

$err[] = "Логин может состоять только из букв английского алфавита и цифр";

}

if(strlen($\_POST['login']) < 3 or strlen($\_POST['login']) > 30)

{

$err[] = "Логин должен быть не меньше 3-х символов и не больше 30";

}

// проверяем, не сущестует ли пользователя с таким именем

$query = mysqli\_query($link, "SELECT user\_id FROM users WHERE user\_login='".mysqli\_real\_escape\_string($link, $\_POST['login'])."'");

if(mysqli\_num\_rows($query) > 0)

{

$err[] = "Пользователь с таким логином уже существует в базе данных";

}

// Если нет ошибок, то добавляем в БД нового пользователя

if(count($err) == 0)

{

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

// Убераем лишние пробелы и делаем двойное хеширование

$password = md5(md5(trim($\_POST['password'])));

mysqli\_query($link,"INSERT INTO users SET user\_login='".$login."', user\_password='".$password."'");

header("Location: login.php"); exit();

}

else

{

print "<b>При регистрации произошли следующие ошибки:</b><br>";

foreach($err AS $error)

{

print $error."<br>";

}

}

}

?>

<form method="POST">

Логин <input name="login" type="text" required><br>

Пароль <input name="password" type="password" required><br>

<input name="submit" type="submit" value="Зарегистрироваться">

</form>

login.php

<?

// Страница авторизации

// Функция для генерации случайной строки

function generateCode($length=6) {

$chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHI JKLMNOPRQSTUVWXYZ0123456789";

$code = "";

$clen = strlen($chars) - 1;

while (strlen($code) < $length) {

$code .= $chars[mt\_rand(0,$clen)];

}

return $code;

}

// Соединямся с БД

$link=mysqli\_connect("localhost", "mysql\_user", "mysql\_password", "testtable");

if(isset($\_POST['submit']))

{

// Вытаскиваем из БД запись, у которой логин равняеться введенному

$query = mysqli\_query($link,"SELECT user\_id, user\_password FROM users WHERE user\_login='".mysqli\_real\_escape\_string($link,$\_POST['login'])."' LIMIT 1");

$data = mysqli\_fetch\_assoc($query);

// Сравниваем пароли

if($data['user\_password'] === md5(md5($\_POST['password'])))

{

// Генерируем случайное число и шифруем его

$hash = md5(generateCode(10));

if(!empty($\_POST['not\_attach\_ip']))

{

// Если пользователя выбрал привязку к IP

// Переводим IP в строку

$insip = ", user\_ip=INET\_ATON('".$\_SERVER['REMOTE\_ADDR']."')";

}

// Записываем в БД новый хеш авторизации и IP

mysqli\_query($link, "UPDATE users SET user\_hash='".$hash."' ".$insip." WHERE user\_id='".$data['user\_id']."'");

// Ставим куки

setcookie("id", $data['user\_id'], time()+60\*60\*24\*30, "/");

setcookie("hash", $hash, time()+60\*60\*24\*30, "/", null, null, true); // httponly !!!

// Переадресовываем браузер на страницу проверки нашего скрипта

header("Location: check.php"); exit();

}

else

{

print "Вы ввели неправильный логин/пароль";

}

}

?>

<form method="POST">

Логин <input name="login" type="text" required><br>

Пароль <input name="password" type="password" required><br>

Не прикреплять к IP(не безопасно) <input type="checkbox" name="not\_attach\_ip"><br>

<input name="submit" type="submit" value="Войти">

</form>

check.php

<?

// Скрипт проверки

// Соединямся с БД

$link=mysqli\_connect("localhost", "mysql\_user", "mysql\_password", "testtable");

if (isset($\_COOKIE['id']) and isset($\_COOKIE['hash']))

{

$query = mysqli\_query($link, "SELECT \*,INET\_NTOA(user\_ip) AS user\_ip FROM users WHERE user\_id = '".intval($\_COOKIE['id'])."' LIMIT 1");

$userdata = mysqli\_fetch\_assoc($query);

if(($userdata['user\_hash'] !== $\_COOKIE['hash']) or ($userdata['user\_id'] !== $\_COOKIE['id'])

or (($userdata['user\_ip'] !== $\_SERVER['REMOTE\_ADDR']) and ($userdata['user\_ip'] !== "0")))

{

setcookie("id", "", time() - 3600\*24\*30\*12, "/");

setcookie("hash", "", time() - 3600\*24\*30\*12, "/", null, null, true); // httponly !!!

print "Хм, что-то не получилось";

}

else

{

print "Привет, ".$userdata['user\_login'].". Всё работает!";

}

}

else

{

print "Включите куки";

}

?>

logout.php

<?

// Страница разавторизации

// Удаляем куки

setcookie("id", "", time() - 3600\*24\*30\*12, "/");

setcookie("hash", "", time() - 3600\*24\*30\*12, "/",null,null,true); // httponly !!!

// Переадресовываем браузер на страницу проверки нашего скрипта

header("Location: /"); exit;

?>

<form method="POST">

Логин <input name="login" type="text" required><br>

Пароль <input name="password" type="password" required><br>

Не прикреплять к IP(не безопасно) <input type="checkbox" name="not\_attach\_ip"><br>

<input name="submit" type="submit" value="Войти">

</form>