Министерство науки и высшего образования РФ

Государственное образовательное учреждение высшего профессионального образования

Белгородский Государственный Технологический Университет им В. Г. Шухова

Кафедра программного обеспечения вычислительной техники и автоматизированных систем.

Расчётно-графическое задание по дисциплине

«Технологии Web-программирования»

Выполнил:
студент группы ПВ-41
Колесников В.Д.
Проверил: Картамышев С. В.

Белгород 2019 г.

Оглавление

Цель работы

Целью данного проекта является изучение принципов работы протокола передачи гипертекста HTTP и языка CSS, ознакомление с основами разработки frontend-приложений, в частности, с использованием клиентских фреймворков (Vue.js), ознакомление с основами работы docker и фреймворком Yii2 и изучить основы взаимодействия web-приложения с базой данных и принципы работы отправки Ajax запросов.

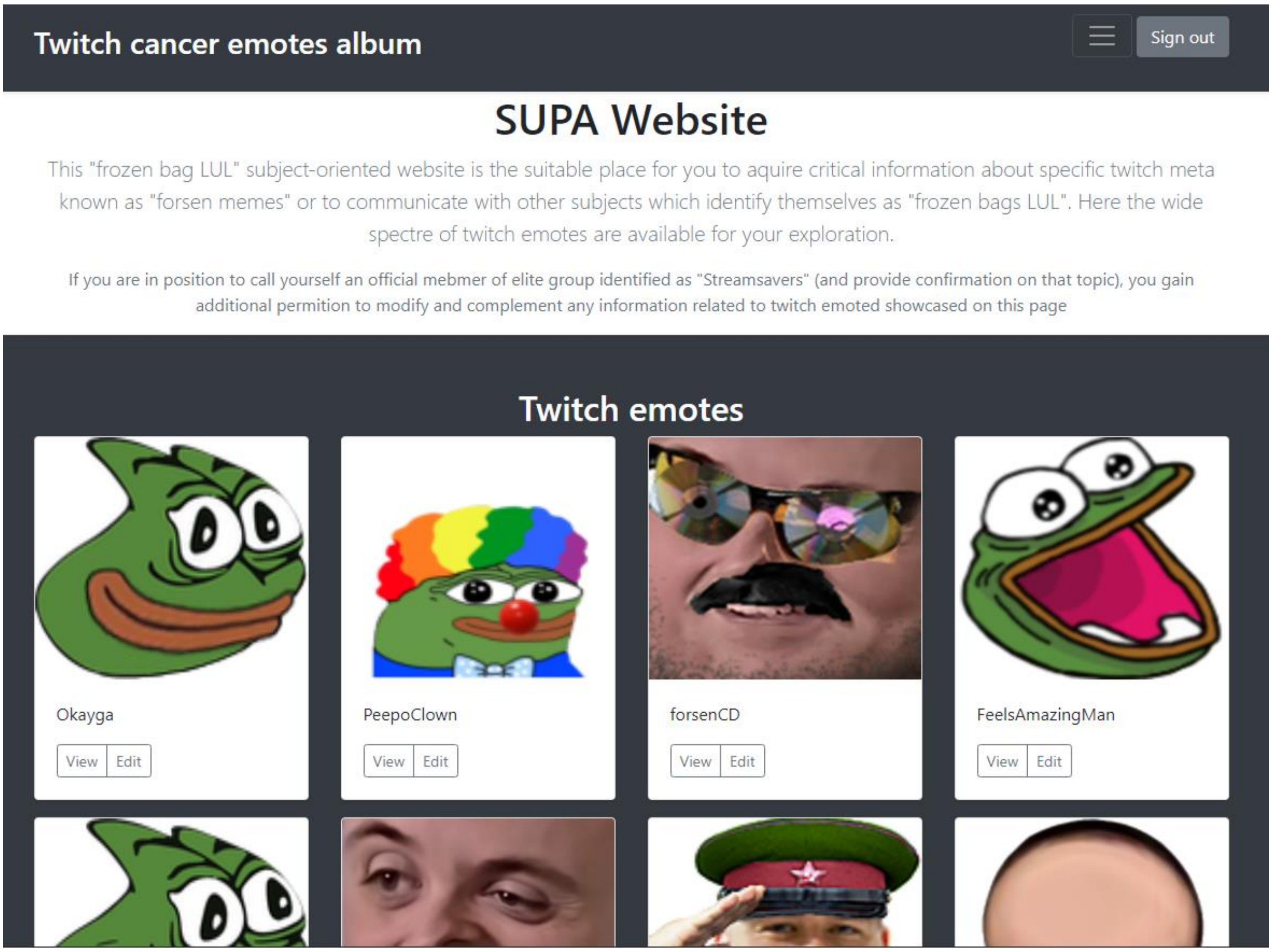
Выполнение

В качестве темы проекта был выбран сайт для интеграции с платформой Twitch. По сути, данный сайт представляет собой альбом различных изображений и эмоций, имеющих название и описание. Также предусмотрена возможность изменения изображений, удаления или добавления новых.

Также имеются пользователи, которые могут заходить на сайт (через систему логина и регистрации) и оставлять комментарии к эмоциям.

Глава 1. HTML страницы

Сначала были спроектированы сами HTML страницы. В качестве основы и макета был выбран макет Album из примеров на сайте Bootstrap, карточки которого будут представлять собой вышеописанные изображения. Немного изменив данный шаблон, получаем следующий результат:



Twitch cancer emotes album

SUPA Website

This "frozen bag LUL" subject-oriented website is the suitable place for you to aquire critical information about specific twitch meta known as "forsen memes" or to communicate with other subjects which identify themselves as "frozen bags LUL". Here the wide spectre of twitch emotes are available for your exploration.

If you are in position to call yourself an official mebmer of elite group identified as "Streamsavers" (and provide confirmation on that topic), you gain additional permition to modify and complement any information related to twitch emoted showcased on this page

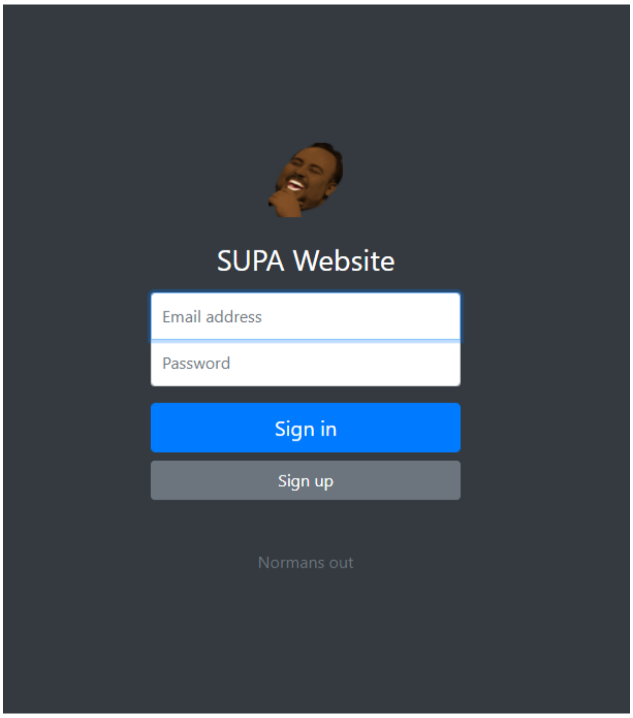
Twitch emotes

Okayga

PeepoClown

forsenCD

FeelsAmazingMan



Код страниц и css классов:

album.html

```
<!doctype html>
<html lang="en">
 <head>
  <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
   <meta name="description" content="">
   <meta name="author" content="Pepega">
   <meta name="generator" content="Jekyll v4.1.1">
   <title>Album example · Bootstrap</title>

   <link rel="canonical" href="https://getbootstrap.com/docs/4.5/examples/album/">
   <!-- Bootstrap core CSS -->
  <link href="../assets/dist/css/bootstrap.min.css" rel="stylesheet">

   <style>
    .bd-placeholder-img {
    font-size: 1.125rem;
    text-anchor: middle;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
    }

    @media (min-width: 768px) {
    .bd-placeholder-img-lg {
     font-size: 3.5rem;
      }
    }
   </style>
   <!-- Custom styles for this template -->
   <link href="album.css" rel="stylesheet">
  </head>
 <header>
  <div class="collapse bg-dark" id="navbarHeader">
   <div class="container">
    <div class="row">
     <div class="col-sm-8 col-md-7 py-4">
      <h4 class="text-white">VI VON ZULUL</h4>
      <p class="text-muted">If you understand this communique, then you have been defeated and now you must
post identical communique in Forsen's chatroom. ZULUL</p>
      <a href="https://www.twitch.tv/forsen" class="btn btn-secondary my-2">Post it ZULUL</a>
     </div>
    </div>
   </div>
  </div>
 <div class="navbar navbar-dark bg-dark shadow-sm">
  <div class="container d-flex justify-content-between">
   <a href="#" class="navbar-brand d-flex align-items-center">
    <h3>Twitch cancer emotes album</h3>
   </a>
   <p>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader" aria-controls="navbarHeader" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
    </button>
    <a href="signin.html" class="btn btn-secondary my-2">Sign out</a>
   </p>
   </div>
  </div>
 </header>

 <main role="main" >
  <section class="text-center">
   <div class="container">
    <h1>SUPA Website</h1>
    <p class="lead text-muted">This "frozen bag LUL" subject-oriented website is the suitable place for you to aquire critical information about specific twitch meta known as "forsen memes" or to communicate with other subjects which identify themselves as "frozen bags LUL". Here the wide spectre of twitch emotes are available for your exploration.</p>
    <p class="text-muted">If you are in position to call yourself an official mebmer of elite group identified as "Streamsavers" (and provide confirmation on that topic), you gain additional permition to modify and complement any information related to twitch emoted showcased on this page</p>
   </div>
  </section>
  <div class="album py-5 bg-dark">
   <div class="container">
    <center>
     <h2 align="text-center"><font color="white">Twitch emotes</font></h2>
    </center>
    <div class="row">
     <div class="col-md-3" >
      <div class="card mb-3 shadow-sm" >
       <img class="bd-placeholder-img card-img-top" src="../assets/images/Okayga.png" width="100%" height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">Okayga</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
     <div class="col-md-3">
      <div class="card mb-3 shadow-sm">
       <img class="bd-placeholder-img card-img-top" src="../assets/images/PeepoClown.png" width="100%" height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">PeepoClown</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
     <div class="col-md-3">
      <div class="card mb-3 shadow-sm">
       <img class="bd-placeholder-img card-img-top" src="../assets/images/forsenCD.jpg" width="100%" height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">forsenCD</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
     <div class="col-md-3">
      <div class="card mb-3 shadow-sm">
       <img class="bd-placeholder-img card-img-top" src="../assets/images/FeelsAmazingMan.png" width="100%"
height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">FeelsAmazingMan</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
     <div class="col-md-3">
      <div class="card mb-3 shadow-sm">
       <img class="bd-placeholder-img card-img-top" src="../assets/images/Pepega.png" width="100%"
height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">Pepega</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
     <div class="col-md-3">
      <div class="card mb-3 shadow-sm">
       <img class="bd-placeholder-img card-img-top" src="../assets/images/forsenE.jpg" width="100%"
height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">forsenE</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
     <div class="col-md-3">
      <div class="card mb-3 shadow-sm">
       <img class="bd-placeholder-img card-img-top" src="../assets/images/KKomrade.png" width="100%"
height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">KKomrade</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
     <div class="col-md-3">
      <div class="card mb-3 shadow-sm">
       <img class="bd-placeholder-img card-img-top" src="../assets/images/5Head.png" width="100%"
height="225" role="img" aria-label="Placeholder: Image"></img>
       <div class="card-body">
        <p class="card-text">5Head</p>
        <div class="d-flex justify-content-between align-items-center">
         <div class="btn-group">
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#viewModal">View</button>
          <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#editModal">Edit</button>
         </div>
        </div>
       </div>
      </div>
     </div>
    </div>
   </div>
   <p class="text-center">
    <button type="button" class="btn btn-secondary" data-toggle="modal" data-target="#addModal">Add new
image</button>
   </p>
  </div>
 </main>

 <div class="modal fade" id="addModal" tabindex="-1" role="dialog" aria-labelledby="addModalLabel" aria-
hidden="true">
  <div class="modal-dialog" role="document">
   <div class="modal-content">
    <div class="modal-header">
     <h5 class="modal-title" id="exampleModalLabel">Add new image</h5>
     <button type="button" class="close" data-dismiss="modal" aria-label="Close">
      <span aria-hidden="true">&times;</span>
     </button>
    </div>
    <div class="modal-body">
     <label for="imageURL">URL</label>
     <input type="text" class="form-control" id="imageURL">
     <label for="imageName">Name</label>
     <input type="text" class="form-control" id="imageName">
     <label for="description">Description</label>
     <textarea type="text" class="form-control" id="description" rows="10"></textarea>
    </div>
    <div class="modal-footer">
     <button type="button" class="btn btn-primary">Add image</button>
    </div>
   </div>
  </div>
 </div>

 <div class="modal fade" id="editModal" tabindex="-1" role="dialog" aria-labelledby="editModalLabel" aria-
hidden="true">
  <div class="modal-dialog" role="document">
   <div class="modal-content">
    <div class="modal-header">
     <h5 class="modal-title" id="exampleModalLabel">Edit image</h5>
     <button type="button" class="close" data-dismiss="modal" aria-label="Close">
      <span aria-hidden="true">&times;</span>
     </button>
    </div>
    <div class="modal-body">
     <label for="imageURL">URL</label>
     <input type="text" class="form-control" id="imageURL">
     <label for="imageName">Name</label>
     <input type="text" class="form-control" id="imageName">
     <label for="description">Description</label>
     <textarea type="text" class="form-control" id="description" rows="10"></textarea>
    </div>
    <div class="modal-footer">
     <button type="button" class="btn btn-secondary" data-toggle="modal" data-
target="#confirmDeleteModal">Delete image</button>
     <button type="button" class="btn btn-primary">Save image</button>
    </div>
   </div>
  </div>
 </div>

 <div class="modal fade" id="viewModal" tabindex="-1" role="dialog" aria-labelledby="viewModalLongTitle"
aria-hidden="true">
  <div class="modal-dialog" role="document">
   <div class="modal-content">
    <div class="modal-header">
     <h5 class="modal-title" id="exampleModalLabel">Okayga</h5>
     <button type="button" class="close" data-dismiss="modal" aria-label="Close">
      <span aria-hidden="true">&times;</span>
     </button>
    </div>
    <div class="modal-body">
     <img class="bd-placeholder-img mb-3 card-img-top" src="../assets/images/Okayga.png" width="100%"
height="100%" role="img" aria-label="Placeholder: Image"></img>
     <p class="lead text">Pepega + FeelsOkayMan. Best emote.</p>
    </div>
   </div>
  </div>
 </div>

 <div class="modal fade" id="confirmDeleteModal" tabindex="-1" role="dialog" aria-
labelledby="confirmDeleteModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-sm" role="document">
   <div class="modal-content">
    <div class="modal-body bg-dark">
     <p class="lead text"><font color="white">Are you sure you want to delete this image?</font></p>
     <button type="button" class="btn btn-primary" data-dismiss="modal">Yes</button>
     <button type="button" class="btn btn-secondary" data-dismiss="modal">No</button>
    </div>
   </div>
  </div>
 </div>

 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
 <script>window.jQuery || document.write('<script
src="../assets/js/vendor/jquery.slim.min.js"><\/script>')</script><script
src="../assets/dist/js/bootstrap.bundle.min.js"></script>
</html>
```

signin.html

```
<!doctype html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
  <meta name="generator" content="Jekyll v4.1.1">
  <title>Signin Template · Bootstrap</title>
  <link rel="canonical" href="https://getbootstrap.com/docs/4.5/examples/sign-in/">
  <!-- Bootstrap core CSS -->
  <link href="../assets/dist/css/bootstrap.min.css" rel="stylesheet">
  <style>
   .bd-placeholder-img {
    font-size: 1.125rem;
    text-anchor: middle;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
   }
   @media (min-width: 768px) {
    .bd-placeholder-img-lg {
     font-size: 3.5rem;
    }
   }
  </style>
  <!-- Custom styles for this template -->
  <link href="signin.css" rel="stylesheet">
 </head>
 <body class="text-center bg-dark">
  <form class="form-signin">
   <img class="mb-4" src="../assets/images/ZULUL.png" alt="" width="72" height="72">
   <h1 class="h3 mb-3 font-weight-normal"><font color="white">SUPA Website</font></h1>
   <label for="inputEmail" class="sr-only">Email address</label>
   <input type="email" id="inputEmail" class="form-control" placeholder="Email address" required autofocus>
   <label for="inputPassword" class="sr-only">Password</label>
   <input type="password" id="inputPassword" class="form-control" placeholder="Password" required>
   <div class="checkbox mb-3"></div>
   <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
   <a href="signup.html" class="btn btn-secondary btn-block">Sign up</a>
   <p class="mt-5 mb-3 text-muted">Normans out</p>
  </form>
 </body>
</html>
```

signup.html

```
<!doctype html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
  <meta name="generator" content="Jekyll v4.1.1">
  <title>Signin Template · Bootstrap</title>
  <link rel="canonical" href="https://getbootstrap.com/docs/4.5/examples/sign-in/">
  <!-- Bootstrap core CSS -->
  <link href="../assets/dist/css/bootstrap.min.css" rel="stylesheet">
  <style>
   .bd-placeholder-img {
    font-size: 1.125rem;
    text-anchor: middle;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
   }
   @media (min-width: 768px) {
    .bd-placeholder-img-lg {
     font-size: 3.5rem;
    }
   }
  </style>
  <!-- Custom styles for this template -->
  <link href="signin.css" rel="stylesheet">
 </head>
 <body class="text-center bg-dark">
  <form class="form-signin">
   <img class="mb-4" src="../assets/images/ZULUL.png" alt="" width="72" height="72">
   <h1 class="h3 mb-3 font-weight-normal"><font color="white">SUPA Website</font></h1>
   <h3 class="h5 mb-3 font-weight-normal"><font color="white">So you want to become a baj...</font></h3>
   <label for="inputEmail" class="sr-only">Enter email address</label>
   <input type="email" id="inputEmail" class="form-control" placeholder="Enter email address">
   <label for="inputPassword" class="sr-only">Create password</label>
   <input type="password" id="inputPassword" class="form-control" placeholder="Create password">
   <div class="checkbox mb-3"></div>
   <button class="btn btn-lg btn-primary btn-block" type="submit">Sign up</button>
   <a href="signin.html" class="btn btn-secondary btn-block">Back</a>
   <p class="mt-5 mb-3 text-muted">FeelsOkayMan</p>
  </form>
 </body>
</html>
```

album.css

```
.jumbotron {
  padding-top: 3rem;
  padding-bottom: 3rem;
  margin-bottom: 0;
  background-color: #fff;
}
@media (min-width: 768px) {
  .jumbotron {
    padding-top: 6rem;
    padding-bottom: 6rem;
  }
}
```

```
.jumbotron p:last-child {
  margin-bottom: 0;
}

.jumbotron h1 {
  font-weight: 300;
}

.jumbotron .container {
  max-width: 40rem;
}

footer {
  padding-top: 3rem;
  padding-bottom: 3rem;
}

footer p {
  margin-bottom: .25rem;

}
```

singin.css

```
html,
body {
  height: 100%;
}

body {
  display: -ms-flexbox;
  display: flex;
  -ms-flex-align: center;
  align-items: center;
  padding-top: 40px;
  padding-bottom: 40px;
  background-color: #f5f5f5;
}

.form-signin {
  width: 100%;
  max-width: 330px;
  padding: 15px;
  margin: auto;
}
.form-signin .checkbox {
  font-weight: 400;
}
.form-signin .form-control {
  position: relative;
  box-sizing: border-box;
  height: auto;
  padding: 10px;
  font-size: 16px;
}
.form-signin .form-control:focus {
  z-index: 2;
}
.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
}
```

Глава 2. Приложение Vue

Далее проект был перенесён на фреймворк vue. Если рассматривать код HTML-файлов, то можно спокойно выделить корточки эмоций в один компонент, так как все карточки одинаковые, и передавать этим компонентам в качестве пропсов данные. В результате получилась следующая структура:

Home.vue

```
<template>
  <div class="home">
    <HomeHeader/>
    <HomeText/>
    <AlbumCards v-bind:emotes="emotes"/>
  </div>
</template>

<script>
// @ is an alias to /src
import HomeHeader from "@/components/home/HomeHeader";
import HomeText from "@/components/home/HomeText";
import AlbumCards from "@/components/home/AlbumCards";

export default {
  name: 'Home',
  components: {
    AlbumCards,
    HomeText,
    HomeHeader
  },
  data(){
    return {
      emotes: [
        {
          id: 0,
          name: 'Okayga',
          description: 'FeelsOkayMan + Pepega.',
          image: require('@/assets/images/Okayga.png'),
        },
        {
          id: 1,
          name: 'PeepoClown',
          description: 'Because twitch viewers are often clowns.',
          image: require('@/assets/images/PeepoClown.png'),
        },
        {
          id: 2,
          name: 'forsenCD',
          description: 'D OMEGALUL C emote.',
          image: require('@/assets/images/forsenCD.jpg'),
        },
        {
          id: 3,
          name: 'FeelsAmazingMan',
          description: 'When you are totally happy.',
          image: require('@/assets/images/FeelsAmazingMan.png'),
        },
        {
          id: 4,
          name: 'Pepega',
          description: 'Absolute retardness.',
          image: require('@/assets/images/Pepega.png'),
        },
        {
          id: 5,
          name: 'forsenE',
          description: 'The face of Twitch.',
          image: require('@/assets/images/forsenE.jpg'),
        },
        {
          id: 6,
          name: 'KKomrade',
          description: 'Fellow russians emote',
          image: require('@/assets/images/KKomrade.png'),
        },
        {
          id: 7,
          name: '5Head',
          description: 'Intellectual.',
          image: require('@/assets/images/5Head.png'),
        },
      ]
    }
  }
}
</script>

<style>

</style>
```

Signin.vue

```
<template>
  <div class="text-center bg-dark">
    <form class="form-signup">
      <img class="mb-4" src="../assets/images/ZULUL.png" alt="" width="72" height="72">
      <h1 class="h3 mb-3 font-weight-normal"><font color="white">SUPA Website</font></h1>
      <label for="inputEmail" class="sr-only">Email address</label>
      <input type="email" id="inputEmail" class="form-control" placeholder="Email address" required
autofocus>
      <label for="inputPassword" class="sr-only">Password</label>
      <input type="password" id="inputPassword" class="form-control" placeholder="Password" required>
      <div class="checkbox mb-3"></div>
      <router-link class="btn btn-lg btn-primary btn-block" v-bind:to="{name: 'Home'}">Sign in</router-link>
      <router-link class="btn btn-secondary btn-block" v-bind:to="{name: 'SignUp'}">Sign up</router-link>
      <p class="mt-5 mb-3 text-muted">Normans out</p>
    </form>
  </div>
</template>

<script>
export default {
  name: "SignIn"
}
</script>

<style scoped>
.form-signup {
  width: 100%;
  max-width: 330px;
  padding: 15px;
  margin: auto;
}
.form-signup .checkbox {
  font-weight: 400;
}
.form-signup .form-control {
  position: relative;
  box-sizing: border-box;
  height: auto;
  padding: 10px;
  font-size: 16px;
}
.form-signup .form-control:focus {
  z-index: 2;
}
.form-signup input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
}
.form-signup input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
}
.bg-dark {
  height: 100%;
  display: -ms-flexbox;
  display: flex;
  -ms-flex-align: center;
  align-items: center;
  padding-top: 40px;
  padding-bottom: 40px;
  background-color: #f5f5f5;
}
</style>
```

Signup.vue

```
<template>
  <div class="text-center bg-dark">
    <form class="form-signin">
      <img class="mb-4" src="../assets/images/ZULUL.png" alt="" width="72" height="72">
      <h1 class="h3 mb-3 font-weight-normal"><font color="white">SUPA Website</font></h1>
      <h3 class="h5 mb-3 font-weight-normal"><font color="white">So you want to become a baj...</font></h3>
      <label for="inputEmail" class="sr-only">Enter email address</label>
      <input type="email" id="inputEmail" class="form-control" placeholder="Enter email address">
      <label for="inputPassword" class="sr-only">Create password</label>
      <input type="password" id="inputPassword" class="form-control" placeholder="Create password">
      <div class="checkbox mb-3"></div>
      <router-link class="btn btn-lg btn-primary btn-block" v-bind:to="{name: 'SignIn'}">Sign up</router-
link>
      <router-link class="btn btn-secondary btn-block" v-bind:to="{name: 'SignIn'}">Back</router-link>
      <p class="mt-5 mb-3 text-muted">FeelsOkayMan</p>
    </form>
  </div>
</template>

<script>
export default {
  name: "LogIn"
}
</script>

<style scoped>
.form-signin {
  width: 100%;
  max-width: 330px;
  padding: 15px;
  margin: auto;
}
.form-signin .checkbox {
  font-weight: 400;
}
.form-signin .form-control {
  position: relative;
  box-sizing: border-box;
  height: auto;
  padding: 10px;
  font-size: 16px;
}
.form-signin .form-control:focus {
  z-index: 2;
}
.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
}
.bg-dark {
  height: 100%;
  display: -ms-flexbox;
  display: flex;
  -ms-flex-align: center;
  align-items: center;
  padding-top: 40px;
  padding-bottom: 40px;
  background-color: #f5f5f5;
}
</style>
```

HomeHeader.vue

```
<template>
  <header>
    <div class="collapse bg-dark" id="navbarHeader">
      <div class="container">
        <div class="row">
```

```
          <div class="col-sm-8 col-md-7 py-4">
            <h4 class="text-white">VI VON ZULUL</h4>
            <p class="text-muted">If you understand this communique, then you have been defeated and now you must post identical communique in Forsen's chatroom. ZULUL</p>
            <a href="https://www.twitch.tv/forsen" class="btn btn-secondary my-2">Post it ZULUL</a>
          </div>
        </div>
      </div>
    </div>
    <div class="navbar navbar-dark bg-dark shadow-sm">
      <div class="container d-flex justify-content-between">
        <a href="#" class="navbar-brand d-flex align-items-center">
          <h3>Twitch cancer emotes album</h3>
        </a>
        <p>
          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader" aria-controls="navbarHeader" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <router-link class="btn btn-secondary my-2" v-bind:to="{name: 'SignIn'}">Sign out</router-link>
        </p>
      </div>
    </div>
  </header>
</template>

<script>
export default {
name: "HomeHeader"
}
</script>

<style scoped>
</style>
```

HomeText.vue

```
<template>
  <section class="text-center">
    <div class="container">
      <h1>SUPA Website</h1>
      <p class="lead text-muted">This "frozen bag LUL" subject-oriented website is the suitable place for you to aquire critical information about specific twitch meta known as "forsen memes" or to communicate with other subjects which identify themselves as "frozen bags LUL". Here the wide spectre of twitch emotes are available for your exploration.</p>
      <p class="text-muted">If you are in position to call yourself an official mebmer of elite group identified as "Streamsavers" (and provide confirmation on that topic), you gain additional permition to modify and complement any information related to twitch emoted showcased on this page</p>
    </div>
  </section>
</template>

<script>
export default {
name: "HomeText"
}
</script>

<style scoped>

</style>
```

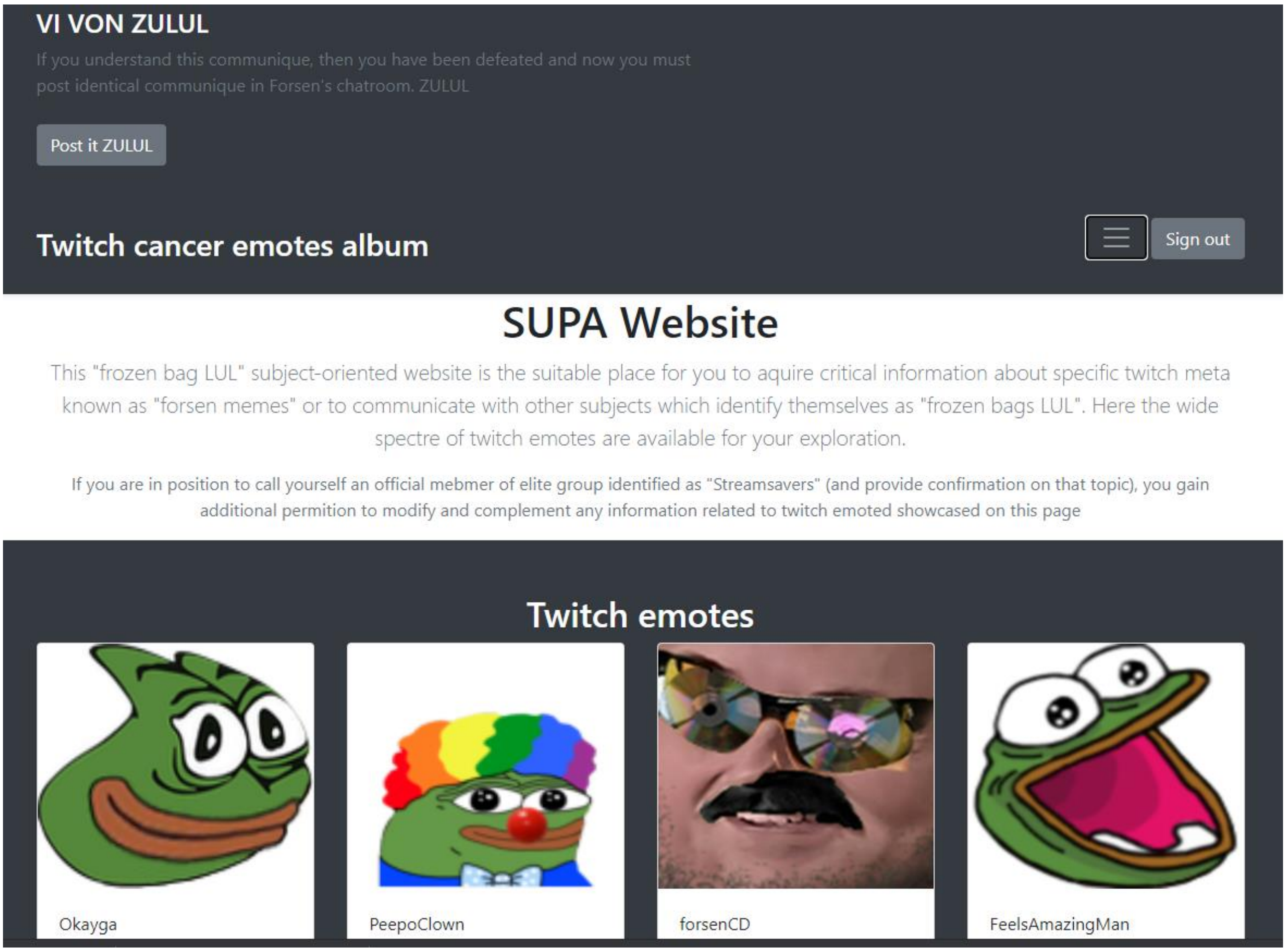
AlbumCards.vue

```
<template>
  <div class="album py-5 bg-dark">
    <div class="container">
      <center>
        <h2 align="text-center"><font color="white">Twitch emotes</font></h2>
      </center>
      <div class="row">
        <AlbumCard v-bind:key="emotesData.id" v-bind:emotes-data="emotesData" v-for="emotesData in emotes" />
      </div>
    </div>
    <p class="text-center">
```

```
      <button type="button" class="btn btn-secondary" data-toggle="modal" data-target="#addModal">Add new
image</button>
    </p>
  </div>
</template>

<script>
import AlbumCard from "@/components/home/AlbumCard";
export default {
name: "AlbumCards",
  components: {
    AlbumCard
  },
  props: [
    'emotes'
  ]

}
</script>

<style scoped>

</style>
```

AlbumCard.vue

```
<template>
  <div class="col-md-3" >
    <div class="card mb-3 shadow-sm" >
      <img v-bind:src="emotesData.image" class="bd-placeholder-img card-img-top" width="100%" height="225"
role="img" aria-label="Placeholder: Image">
      <div class="card-body">
        <p class="card-text">{{emotesData.name}}</p>
        <div class="d-flex justify-content-between align-items-center">
          <div class="btn-group">
            <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#viewModal">View</button>
            <button type="button" class="btn btn-sm btn-outline-secondary" data-toggle="modal" data-
target="#editModal">Edit</button>
          </div>
        </div>
      </div>
    </div>
  </div>
</template>

<script>
export default {
name: "AlbumCard",
  props: [
    'emotesData'
  ]
}
</script>

<style scoped>

</style>
```



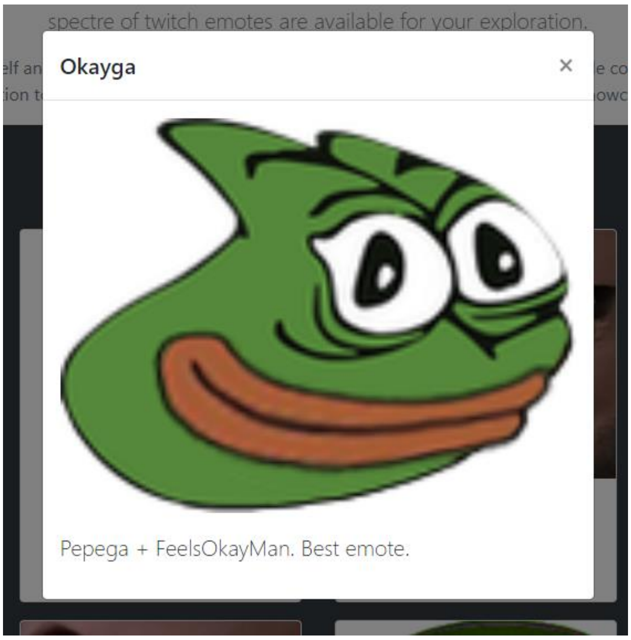
VI VON ZULUL

If you understand this communique, then you have been defeated and now you must post identical communique in Forsen's chatroom. ZULUL

SUPA Website

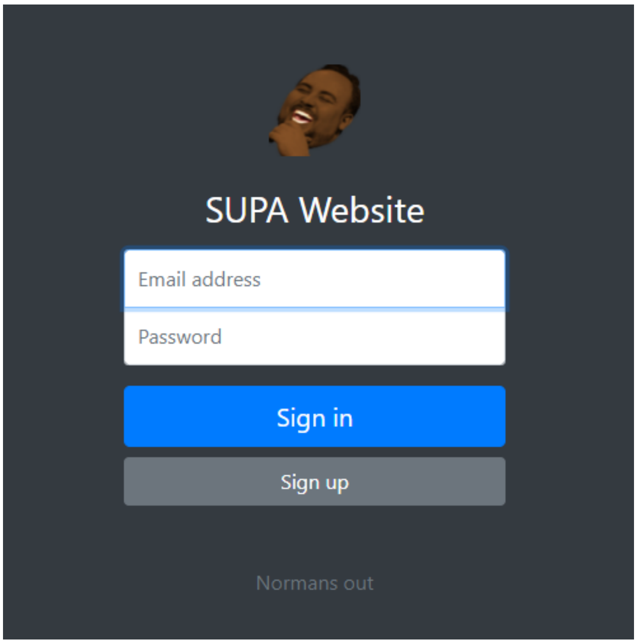
This "frozen bag LUL" subject-oriented website is the suitable place for you to aquire critical information about specific twitch meta known as "forsen memes" or to communicate with other subjects which identify themselves as "frozen bags LUL". Here the wide spectre of twitch emotes are available for your exploration.

If you are in position to call yourself an official mebmer of elite group identified as "Streamsavers" (and provide confirmation on that topic), you gain additional permition to modify and complement any information related to twitch emoted showcased on this page



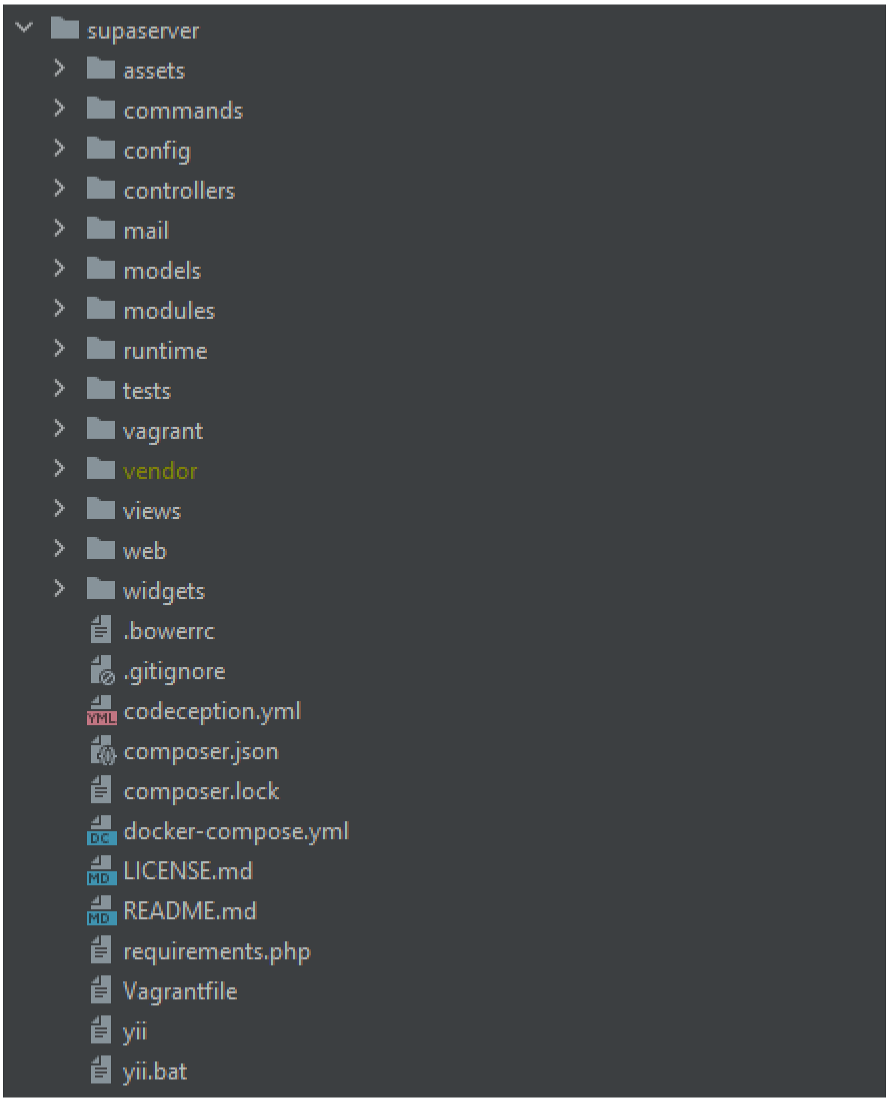
spectre of twitch emotes are available for your exploration.

Pepega + FeelsOkayMan. Best emote.



Глава 3. Серверная часть. Приложение Yii2

Для серверной части сайта было развёрнуто и собрано приложение Yii2 с помощью php-композера командой composer create-project --prefer-dist yiisoft/yii2-app-basic basic.



```
supaserver
  assets
  commands
  config
  controllers
  mail
  models
  modules
  runtime
  tests
  vagrant
  vendor
  views
  web
  widgets
  .bowerrc
  .gitignore
  codeception.yml
  composer.json
  composer.lock
  docker-compose.yml
  LICENSE.md
  README.md
  requirements.php
  Vagrantfile
  yii
  yii.bat
```

Далее был настроен файл конфигурации docker-compose.yml

```
version: '3.1'
services:
  php:
    image: yiisoftware/yii2-php:7.3-apache
    volumes:
      - ~/.composer-docker/cache:/root/.composer/cache:delegated
      - ./:/app:delegated
    ports:
      - '1199:80'
    extra_hosts:
      - "visiticelandbeauty.com:127.0.0.1"

  mysql:
    image: mysql
    command: --default-authentication-plugin=mysql_native_password --character-set-server=utf8 --collation-
server=utf8_general_ci
    environment:
      - MYSQL_ROOT_PASSWORD=verysecret
      - MYSQL_DATABASE=db
      - MYSQL_USER=user
      - MYSQL_PASSWORD=password
```

Был добавлен API модуль версии 1 со следующими контроллерами и моделями:

User.php

```
<?php

namespace app\models;
```

```
class User extends \yii\base\BaseObject implements \yii\web\IdentityInterface
{
    public $id;
    public $username;
    public $password;
    public $authKey;
    public $accessToken;

    private static $users = [
        '100' => [
            'id' => '100',
            'username' => 'admin',
            'password' => 'admin',
            'authKey' => 'test100key',
            'accessToken' => '100-token',
        ],
        '101' => [
            'id' => '101',
            'username' => 'demo',
            'password' => 'demo',
            'authKey' => 'test101key',
            'accessToken' => '101-token',
        ],
    ];


    /**
     * {@inheritdoc}
     */
    public static function findIdentity($id)
    {
        return isset(self::$users[$id]) ? new static(self::$users[$id]) : null;
    }

    /**
     * {@inheritdoc}
     */
    public static function findIdentityByAccessToken($token, $type = null)
    {
        foreach (self::$users as $user) {
            if ($user['accessToken'] === $token) {
                return new static($user);
            }
        }

        return null;
    }

    /**
     * Finds user by username
     *
     * @param string $username
     * @return static|null
     */
    public static function findByUsername($username)
    {
        foreach (self::$users as $user) {
            if (strcasecmp($user['username'], $username) === 0) {
                return new static($user);
            }
        }

        return null;
    }

    /**
     * {@inheritdoc}
     */
    public function getId()
    {
        return $this->id;
    }

    /**
     * {@inheritdoc}
     */
    public function getAuthKey()
    {
        return $this->authKey;
    }

    /**
     * {@inheritdoc}
     */
    public function validateAuthKey($authKey)
    {
        return $this->authKey === $authKey;
    }

    /**
     * Validates password
     *
     * @param string $password password to validate
     * @return bool if password provided is valid for current user
     */
    public function validatePassword($password)
    {
        return $this->password === $password;
    }
}
```

ApiController.php

```
<?php

namespace app\modules\v1\controllers;

use yii\filters\auth\CompositeAuth;
use yii\filters\ContentNegotiator;
use yii\filters\Cors;
use yii\filters\RateLimiter;
use yii\filters\VerbFilter;
use yii\rest\Controller;
use yii\web\Response;

class ApiController extends Controller
{
 public function behaviors()
 {
  return [
   'contentNegotiator' => [
    'class' => ContentNegotiator::className(),
    'formats' => [
     'application/json' => Response::FORMAT_JSON
    ],
   ],
   'verbFilter' => [
    'class' => VerbFilter::className(),
    'actions' => $this->verbs(),
   ],
   'authenticator' => [
    'class' => CompositeAuth::className(),
   ],
   'rateLimiter' => [
    'class' => RateLimiter::className(),
   ],
   'corsFilter' => [
    'class' => Cors::class
   ]
  ];
 }
}
```

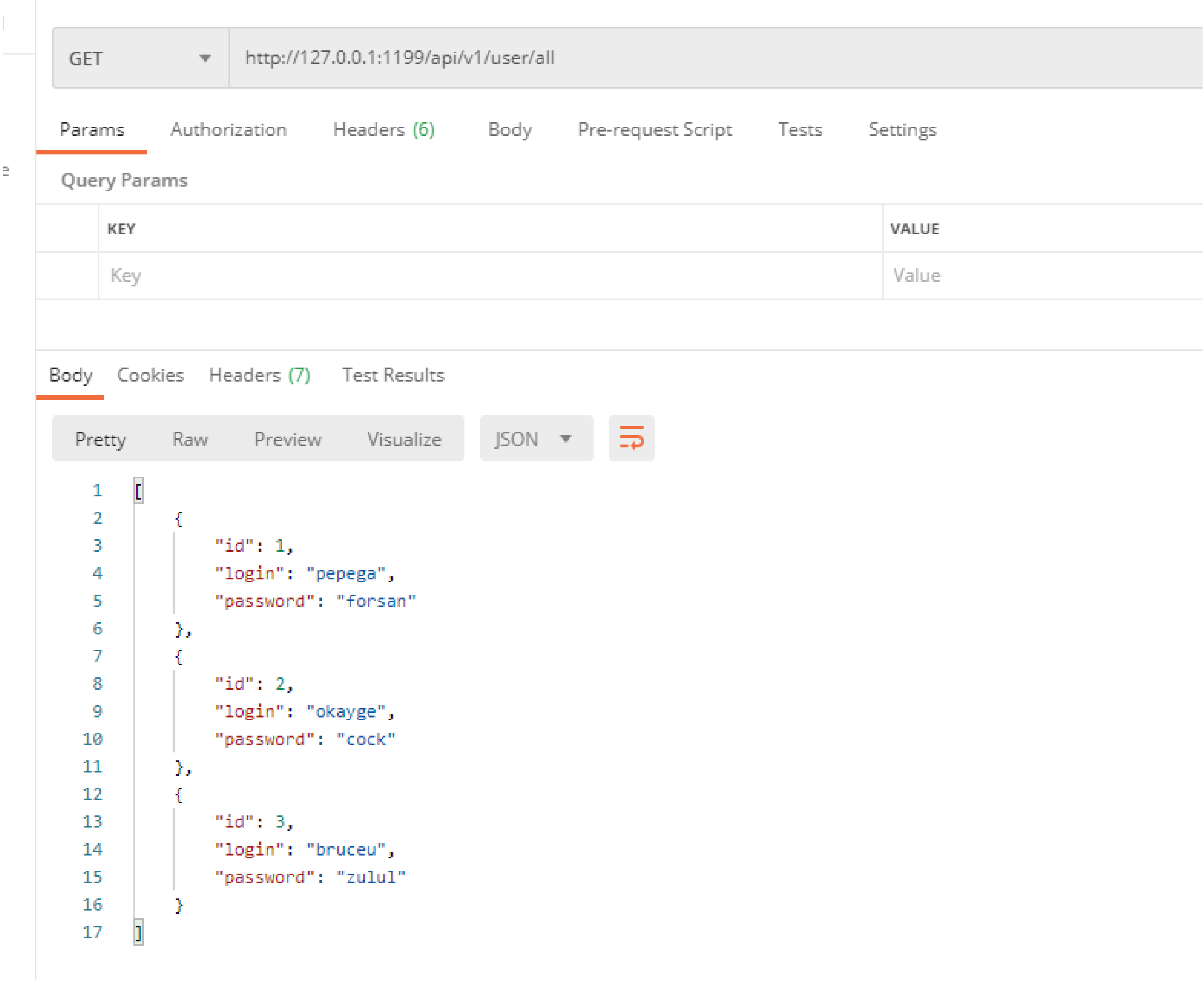
EmotesController.php

```
<?php

namespace app\modules\v1\controllers;

class EmotesController extends ApiController {
 public function actionHome() {
  $emotes = [
   [
    'id'=> 1,
    'name' => 'Okayga',
    'description' => 'FeelsOkayMan + Pepega.',
    'image' => '@/assets/images/Okayga.png',
   ],
   [
    'id' => 2,
    'name' => 'PeepoClown',
    'description' => 'Because twitch viewers are often clowns.',
    'image' => '@/assets/images/PeepoClown.png',
   ],
   [
    'id' => 3,
    'name' => 'forsenCD',
    'description' => 'D OMEGALUL C emote.',
    'image' => '@/assets/images/forsenCD.jpg',
   ],
   [
    'id' => 4,
    'name' => 'FeelsAmazingMan',
    'description' => 'When you are totally happy.',
    'image'=> '@/assets/images/FeelsAmazingMan.png',
   ],
   [
    'id' => 5,
    'name' => 'Pepega',
    'description' => 'Absolute retardness.',
    'image' => '@/assets/images/Pepega.png',
   ],
   [
    'id' => 6,
    'name' => 'forsenE',
    'description' => 'The face of Twitch.',
    'image' => '@/assets/images/forsenE.jpg',
   ],
   [
    'id' => 7,
    'name' => 'KKomrade',
    'description' => 'Fellow russians emote',
    'image' => '@/assets/images/KKomrade.png',
   ],
   [
    'id' => 7,
    'name' => '5Head',
    'description' => 'Intellectual.',
    'image' => '@/assets/images/5Head.png',
   ]
  ];
  return $emotes;
 }
}
```

UserController.php

```
<?php

namespace app\modules\v1\controllers;

use app\modules\v1\models\User;
use yii\rest\Controller;

class UserController extends ApiController {
    public function actionLogin() {
        // login stuff
    }

    public function actionRegister() {
        // register stuff
    }

    public function actionAll() {
        $user = new User();
        return $user -> getUsers();
    }
}
```

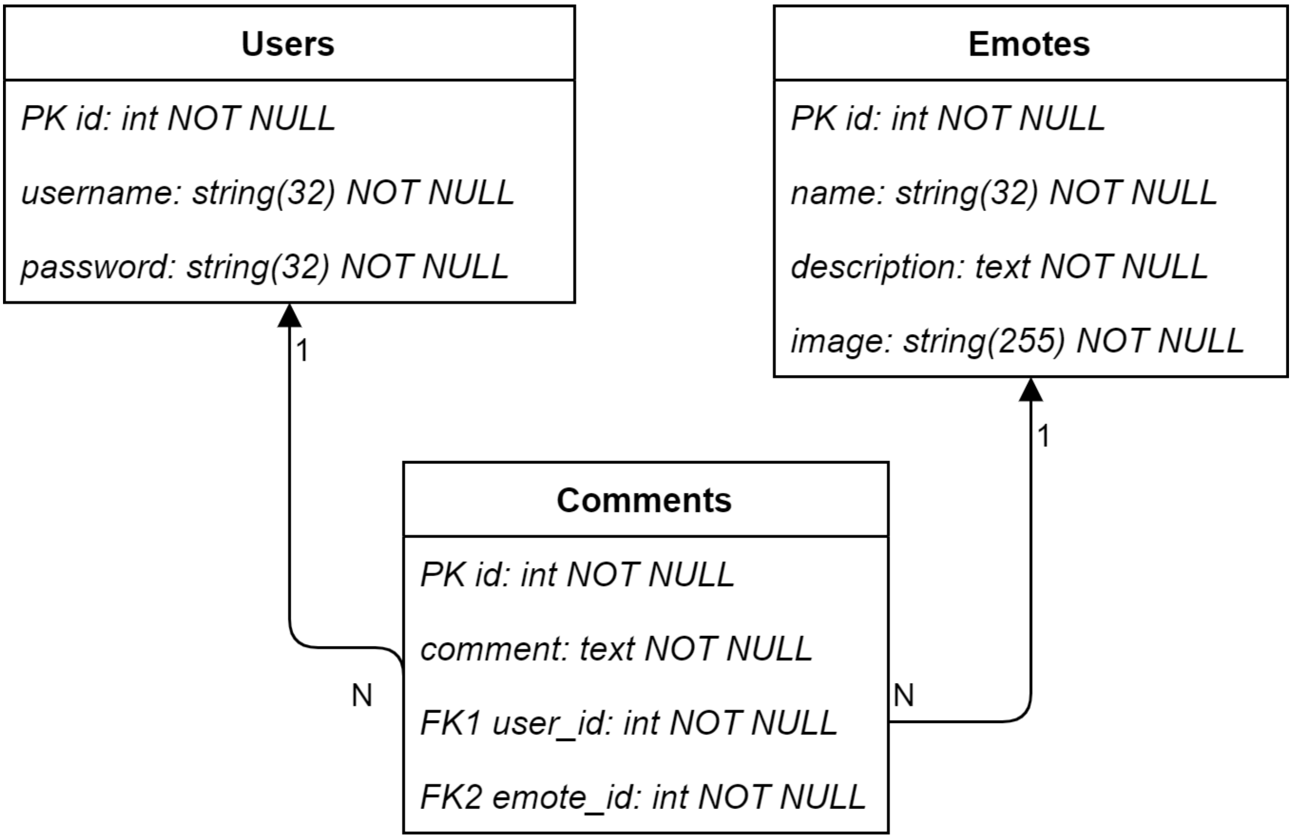
Для удостоверения в корректной работе приложения были выполнены запросы к серверу через Postman.



```
[
    {
        "id": 1,
        "login": "pepega",
        "password": "forsan"
    },
    {
        "id": 2,
        "login": "okayge",
        "password": "cock"
    },
    {
        "id": 3,
        "login": "bruceu",
        "password": "zulul"
    }
]
```

Глава 4. База данных. Миграции и модели.

Далее была разработана база данных, получившая в итоге следующую структуру:


Users
PK id: int NOT NULL
username: string(32) NOT NULL
password: string(32) NOT NULL
Emotes
PK id: int NOT NULL
name: string(32) NOT NULL
description: text NOT NULL
image: string(255) NOT NULL
1
1
Comments
PK id: int NOT NULL
comment: text NOT NULL
FK1 user_id: int NOT NULL
FK2 emote_id: int NOT NULL
N
N

Для реализации данной структуры были созданы миграции:

create_comments_table

```
<?php

use yii\db\Migration;

/**
 * Handles the creation of table `{{%comments}}`.
 */
class m201204_150646_create_comments_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%comments}}', [
            'id' => $this->primaryKey(),
            'comment' => $this->text()->notNull()->comment('Comment text'),
            'createdAt' => $this->dateTime()->notNull()->comment('Creation time'),
            'updatedAt' => $this->dateTime()->notNull()->comment('Update time')
        ]);
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropForeignKey('fk_comments_user_id', '{{%comments}}');
        $this->dropColumn('{{%comments}}', 'user_id');
        $this->dropTable('{{%comments}}');
    }
}
```

create_users_table

```
<?php

use yii\db\Migration;

/**
 * Handles the creation of table `{{%users}}`.
 */
class m201204_151307_create_users_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%users}}', [
            'id' => $this->primaryKey(),
            'username' => $this->string(32)->notNull()->comment('Username'),
            'password' => $this->string(32)->notNull()->comment('Password'),
            'createdAt' => $this->dateTime()->notNull()->comment('Creation time'),
            'updatedAt' => $this->dateTime()->comment('Update time')
        ]);
        $this->addColumn('{{%comments}}', 'user_id', $this->integer()->comment('User'));
        $this->addForeignKey('fk_comments_user_id', '{{%comments}}', 'user_id', '{{%users}}', 'id', 'SET
NULL');
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropTable('{{%users}}');
    }
}
```

create_emotes_table

```
<?php

use yii\db\Migration;

/**
 * Handles the creation of table `{{%emotes}}`.
 */
class m201204_151413_create_emotes_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%emotes}}', [
            'id' => $this->primaryKey(),
            'name' => $this->string(32)->notNull()->comment('Name'),
            'description' => $this->text()->notNull()->comment('Description'),
            'image' => $this->string()->notNull()->comment('Image'),
            'createdAt' => $this->dateTime()->notNull()->comment('Creation time'),
            'updatedAt' => $this->dateTime()->comment('Update time')
        ]);
        $this->addColumn('{{%comments}}', 'emote_id', $this->integer()->comment('Emote'));
        $this->addForeignKey('fk_comments_emote_id', '{{%comments}}', 'emote_id', '{{%emotes}}', 'id', 'SET
NULL');
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropForeignKey('fk_comments_emote_id', '{{%comments}}');
        $this->dropColumn('{{%comments}}', 'emote_id');
        $this->dropTable('{{%emotes}}');
    }
}
```

Далее по созданным миграциям были сгенерированы модели:

BaseModel

```
<?php

namespace app\modules\v1\models;

use yii\behaviors\TimestampBehavior;
use yii\db\ActiveRecord;
use yii\db\Expression;

class BaseModel extends ActiveRecord {
    /**
     * {@inheritdoc}
     */
    public function behaviors() {
        return [
          [
              'class' => TimestampBehavior::class,
              'createdAtAttribute' => 'createdAt',
              'updatedAtAttribute' => 'updatedAt',
              'value' => new Expression('now()')
          ]
        ];
    }
}
```

Comment

```
<?php

namespace app\modules\v1\models;

use Yii;

/**
 * This is the model class for table "comments".
 *
 * @property int $id
 * @property string $comment Comment text
 * @property string $createdAt Creation time
 * @property string $updatedAt Update time
 * @property int|null $user_id User
 * @property int|null $emote_id Emote
 *
 * @property Users $user
 * @property Emotes $emote
 */
class Comment extends BaseModel
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'comments';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['comment', 'createdAt', 'updatedAt'], 'required'],
            [['comment'], 'string'],
            [['createdAt', 'updatedAt'], 'safe'],
            [['user_id', 'emote_id'], 'integer'],
            [['user_id'], 'exist', 'skipOnError' => true, 'targetClass' => Users::className(),
'targetAttribute' => ['user_id' => 'id']],
            [['emote_id'], 'exist', 'skipOnError' => true, 'targetClass' => Emotes::className(),
'targetAttribute' => ['emote_id' => 'id']],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'comment' => 'Comment text',
            'createdAt' => 'Creation time',
            'updatedAt' => 'Update time',
            'user_id' => 'User',
            'emote_id' => 'Emote',
        ];
    }

    /**
     * Gets query for [[User]].
     *
     * @return \yii\db\ActiveQuery
     */
    public function getUser()
    {
        return $this->hasOne(Users::className(), ['id' => 'user_id']);
    }

    /**
     * Gets query for [[Emote]].
     *
     * @return \yii\db\ActiveQuery
     */
    public function getEmote()
    {
        return $this->hasOne(Emotes::className(), ['id' => 'emote_id']);
    }
}
```

User

```
<?php

namespace app\modules\v1\models;

use Yii;

/**
 * This is the model class for table "users".
 *
 * @property int $id
 * @property string $username Username
 * @property string $password Password
 * @property string $createdAt Creation time
 * @property string|null $updatedAt Update time
 *
 * @property Comments[] $comments
 */
```

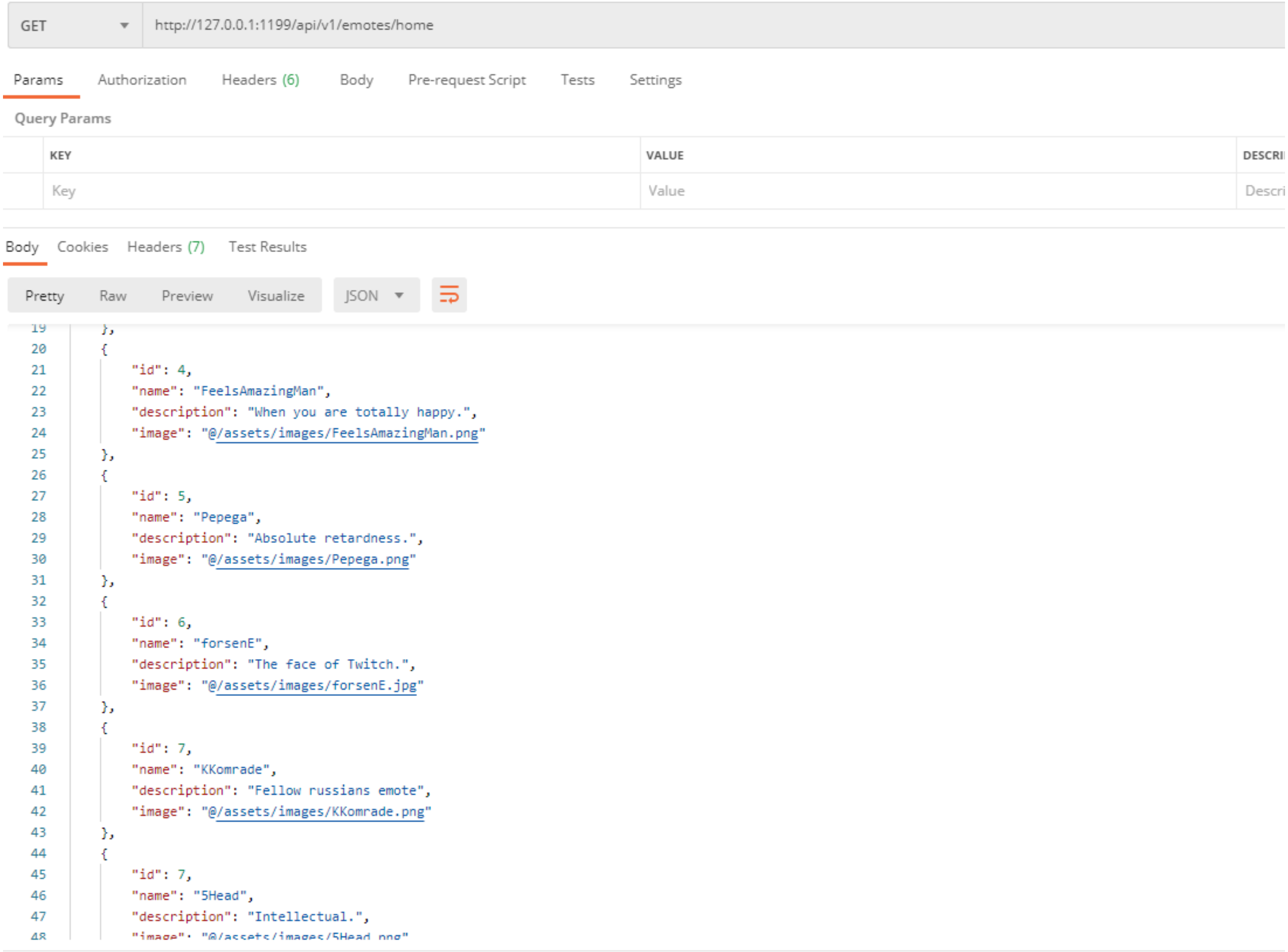
```
class User extends BaseModel
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'users';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['username', 'password', 'createdAt'], 'required'],
            [['createdAt', 'updatedAt'], 'safe'],
            [['username', 'password'], 'string', 'max' => 32],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'username' => 'Username',
            'password' => 'Password',
            'createdAt' => 'Creation time',
            'updatedAt' => 'Update time',
        ];
    }

    /**
     * Gets query for [[Comments]].
     *
     * @return \yii\db\ActiveQuery
     */
    public function getComments()
    {
        return $this->hasMany(Comments::className(), ['user_id' => 'id']);
    }
}
```

Emote

```
<?php

namespace app\modules\v1\models;

use Yii;

/**
 * This is the model class for table "emotes".
 *
 * @property int $id
 * @property string $name Name
 * @property string $description Description
 * @property string $image Image
 * @property string $createdAt Creation time
 * @property string|null $updatedAt Update time
 *
 * @property Comments[] $comments
 */
class Emote extends BaseModel
```

```
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'emotes';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['name', 'description', 'image', 'createdAt'], 'required'],
            [['description'], 'string'],
            [['createdAt', 'updatedAt'], 'safe'],
            [['name'], 'string', 'max' => 32],
            [['image'], 'string', 'max' => 255],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'name' => 'Name',
            'description' => 'Description',
            'image' => 'Image',
            'createdAt' => 'Creation time',
            'updatedAt' => 'Update time',
        ];
    }

    /**
     * Gets query for [[Comments]].
     *
     * @return \yii\db\ActiveQuery
     */
    public function getComments()
    {
        return $this->hasMany(Comments::className(), ['emote_id' => 'id']);
    }
}
```

Для проверки работоспособности был использован Postman.



GET http://127.0.0.1:1199/api/v1/emotes/home

```
    },
    {
        "id": 4,
        "name": "FeelsAmazingMan",
        "description": "When you are totally happy.",
        "image": "@/assets/images/FeelsAmazingMan.png"
    },
    {
        "id": 5,
        "name": "Pepega",
        "description": "Absolute retardness.",
        "image": "@/assets/images/Pepega.png"
    },
    {
        "id": 6,
        "name": "forsenE",
        "description": "The face of Twitch.",
        "image": "@/assets/images/forsenE.jpg"
    },
    {
        "id": 7,
        "name": "KKomrade",
        "description": "Fellow russians emote",
        "image": "@/assets/images/KKomrade.png"
    },
    {
        "id": 7,
        "name": "5Head",
        "description": "Intellectual.",
        "image": "@/assets/images/5Head.png"
```

Глава 5. REST API

Для взаимодействия с базой данных было реализовано следующее REST API:

User

1. Получение списка всех пользователей

URL: /user/all

Тип запроса: GET

Параметры: нет.

2. Логин

URL: /user/login

Тип запроса: POST

Параметры: username (обязательно, логин) , password (обязательно, пароль).

3. Регистрация

URL: /user/register

Тип запроса: POST

Параметры: username (обязательно, логин) , password (обязательно, пароль).

Emote
1. Получение списка всех эмоций
URL: /emote/all
Тип запроса: GET
Параметры: нет.

2. Получение конкретной эмоции
URL: /emote/concrete
Тип запроса: GET
Параметры: id (обязательно, идентификатор объекта).

3. Создание эмоции
URL: /emote/create
Тип запроса: POST
Параметры: принимается JSON-объект с полями: name (обязательно, название эмоции), description (обязательно, описание), image (обязательно, путь\ссылка на изображение).

4. Обновление эмоции
URL: /emote/update
Тип запроса: UPDATE
Параметры: id (обязательно, идентификатор объекта); принимается JSON-объект с полями: name (обязательно, название эмоции), description (обязательно, описание), image (обязательно, путь\ссылка на изображение).

5. Удаление эмоции
URL: /emote/delete
Тип запроса: DELETE
Параметры: id (обязательно, идентификатор объекта).

Comment
1. Получение списка всех комментариев
URL: /comment/all
Тип запроса: GET
Параметры: нет.

2. Получение конкретного комментария по id эмоции, к которой он относится
URL: /comment/emoteid
Тип запроса: GET

Параметры: emote_id (обязательно, идентификатор эмоции).

3. Получение конкретного комментария по id пользователя, к которому он относится

URL: /comment/userid

Тип запроса: GET

Параметры: user_id (обязательно, идентификатор пользователя).

4. Создание комментария

URL: /comment/update

Тип запроса: POST

Параметры: принимается JSON-объект с полями: name (обязательно, название эмоции), description (обязательно, описание), image (обязательно, путь\ссылка на изображение).

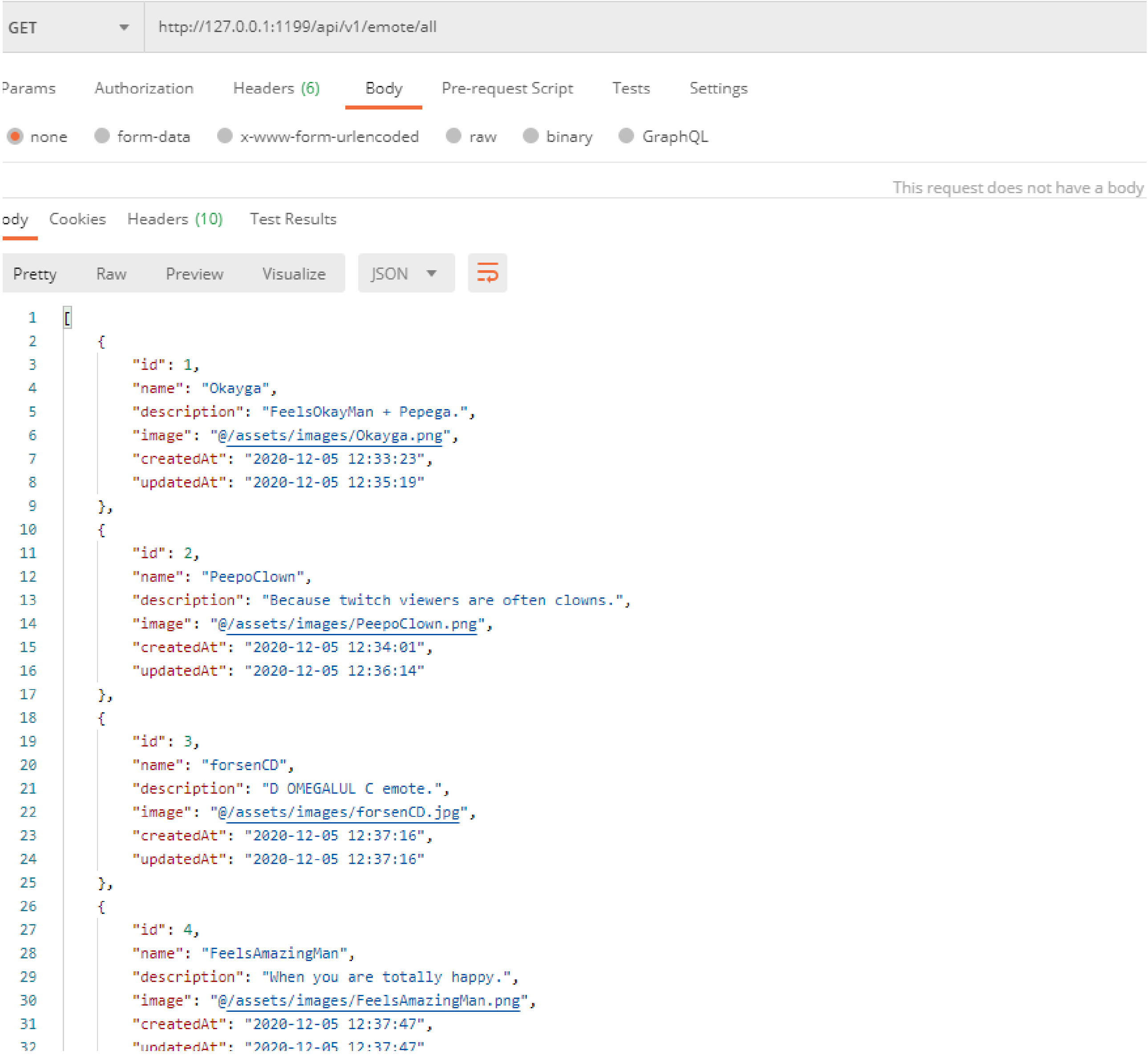
5. Удаление комментария

URL: /comment/delete

Тип запроса: DELETE

Параметры: id (обязательно, идентификатор объекта).

Для проверки работоспособности был использован Postman.



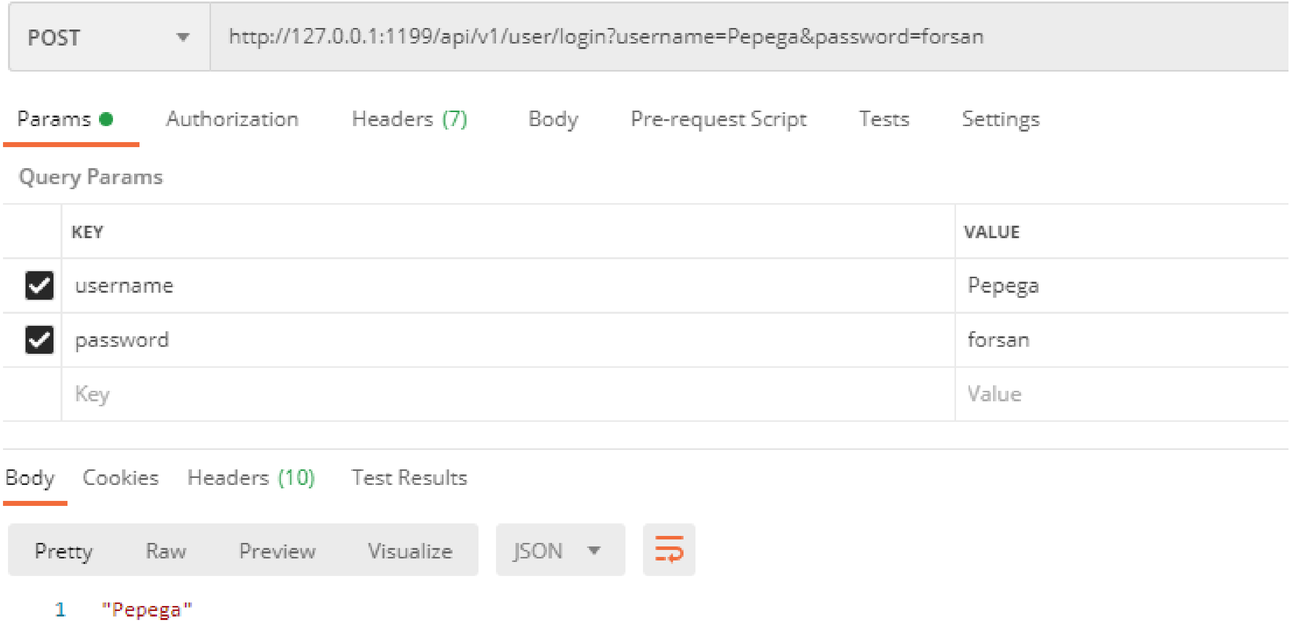
GET http://127.0.0.1:1199/api/v1/emote/all

Body: ● none

This request does not have a body

JSON

```
[
    {
        "id": 1,
        "name": "Okayga",
        "description": "FeelsOkayMan + Pepega.",
        "image": "@/assets/images/Okayga.png",
        "createdAt": "2020-12-05 12:33:23",
        "updatedAt": "2020-12-05 12:35:19"
    },
    {
        "id": 2,
        "name": "PeepoClown",
        "description": "Because twitch viewers are often clowns.",
        "image": "@/assets/images/PeepoClown.png",
        "createdAt": "2020-12-05 12:34:01",
        "updatedAt": "2020-12-05 12:36:14"
    },
    {
        "id": 3,
        "name": "forsenCD",
        "description": "D OMEGALUL C emote.",
        "image": "@/assets/images/forsenCD.jpg",
        "createdAt": "2020-12-05 12:37:16",
        "updatedAt": "2020-12-05 12:37:16"
    },
    {
        "id": 4,
        "name": "FeelsAmazingMan",
        "description": "When you are totally happy.",
        "image": "@/assets/images/FeelsAmazingMan.png",
        "createdAt": "2020-12-05 12:37:47",
        "updatedAt": "2020-12-05 12:37:47"
```



http://127.0.0.1:1199/api/v1/user/login?username=Pepega&password=forsan

	KEY	VALUE
☑	username	Pepega
☑	password	forsan
	Key	Value

```
"Pepega"
```

Глава 6. Взаимодействие приложения с API

Вся логика взаимодействия с REST API была реализована в ранее созданном приложении Vue:

Регистрация, вход и выход:

SignIn.vue

```
import router from "../router";

export default {
  name: "SignIn",
  data() {
    return {
      username: '',
      password: '',
    }
  },
  methods: {
    handleSubmit: function (e) {
      e.preventDefault()
      const submissionData = {
        username: this.username,
        password: this.password,
      }
      this.$http.post('/user/login', submissionData).then((response) => {
        localStorage.setItem('userData', JSON.stringify(response.data))
        router.push({ name: 'Home' })
      }).catch((error) => alert(error.response.data.statusText))
    }
  }
}
```

SignUp.vue

```
import router from "../router";

export default {
  name: "SignUp",
  data() {
    return {
      username: '',
      password: '',
    }
  },
  methods: {
    handleSubmit: function (e) {
      e.preventDefault()
      const submissionData = {
        username: this.username,
        password: this.password,
      }
      this.$http.post('/user/register', submissionData).then((response) => {
        localStorage.setItem('userData', JSON.stringify(response.data))
        router.push({ name: 'SignIn' })
      }).catch((error) => alert(error.response.data.statusText))
    }
  }
}
```

Работа с эмоциями:

Home.vue

```
createEmote(submissionData) {
  const userData = JSON.parse(localStorage.getItem('userData'));
  this.$http.post('/emote/create', submissionData, {params: {accessToken: userData.accessToken}}).then(() =>
{
    this.$http.get('/emote/all').then((response) => this.emotes = [...response.data])
  }).catch((error) => alert(error.response.data.statusText))
},
deleteEmote(emoteID){
  const userData = JSON.parse(localStorage.getItem('userData'));
  this.$http.delete('/emote/delete', { params: { emoteID, accessToken: userData.accessToken } }).then(() => {
    this.$http.get('/emote/all').then((response) => this.emotes = [...response.data])
  }).catch((error) => alert(error.response.data.statusText))
},
editEmote(emoteID, submissionData){
  const userData = JSON.parse(localStorage.getItem('userData'));
  this.$http.post('/emote/update', submissionData, { params: { emoteID, accessToken: userData.accessToken }
}).then(() => {
    this.$http.get('/emote/all').then((response) => this.emotes = [...response.data])
  }).catch((error) => alert(error.response.data.statusText))
}
```

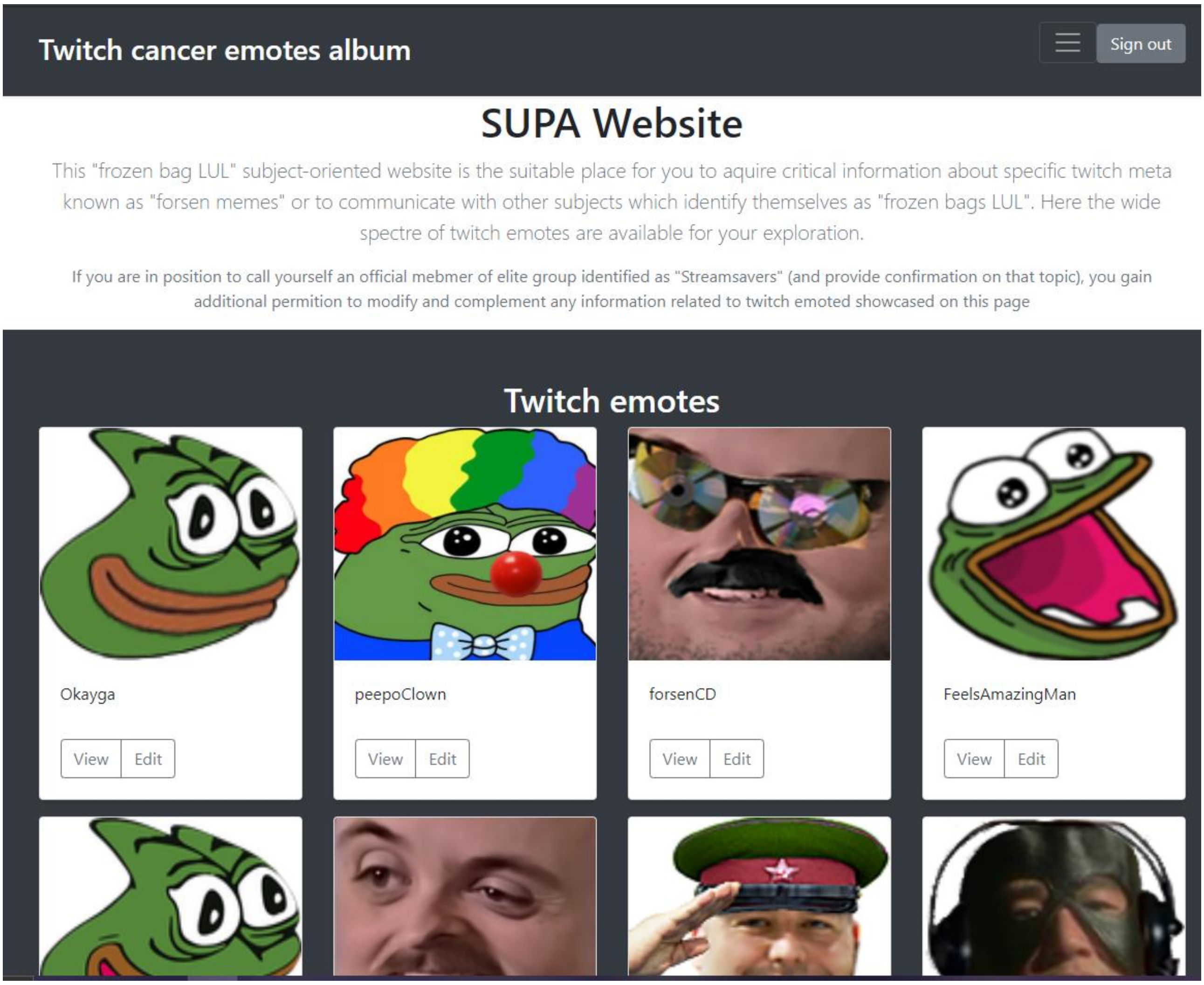
Работа с комментариями:

AlbumCard.vue

```
   import Comment from "./Comment";
   export default {
     name: "AlbumCard",
     data(){
       return{
         comments: [],
         users: [],
         emote: null,
         userData: null,
         newComment: '',
         url: '',
         name: '',
         description: ''
       }
     },
     components: {
       Comment
     },
     props: [
       'emotesData'
     ],
     created(){
       this.emote = this.emotesData
       this.$http.get('/comment/emoteid', { params: { emote_id: this.emote.id } })
           .then((response) => this.comments = [...response.data])
       this.$http.get('/user/all')
           .then((response) => this.users = [...response.data])
       if (localStorage.getItem('userData') === null) {
         return
       }
       const userData = JSON.parse(localStorage.getItem('userData'));
       this.userData = { ...userData }
     },
     methods: {
       viewModal() {
         this.$refs['viewModal'].show()
       },
       editModal() {
         this.$refs['editModal'].show()
       },
       saveModal() {
         if (localStorage.getItem('userData') === null) {
           alert('User not authorised.')
           return
         }
         const submissionData = {
           name: this.name,
           description: this.description,
           image: this.url
         }
         this.$emit('edit-emote', this.emotesData.id, submissionData)
         this.name = ''
         this.description = ''
         this.url = ''
         this.$refs['editModal'].hide()
       },
       deleteModal(emoteID) {
         if (localStorage.getItem('userData') === null) {
           alert('User not authorised.')
           return
         }
         this.$emit('delete-emote', emoteID)
         this.$refs['editModal'].hide()
       },
       submitComment(){
         if (this.newComment !== '') {
           if (localStorage.getItem('userData') === null) {
             alert('User not authorised.')
             return
           }
           const userData = JSON.parse(localStorage.getItem('userData'));
           const submissionData = {
             comment: this.newComment,
             user_id: userData.id,
             emote_id: this.emote.id
           }
           console.log(submissionData)
           console.log(userData.accessToken)
           this.$http.post('/comment/create', submissionData, { params: { accessToken: userData.accessToken }
}).then(() => {
             this.$http.get('/comment/emoteid', { params: { emote_id: this.emote.id } }).then((response) =>
this.comments = [...response.data])
           }).catch((error) => alert(error.response.data.statusText))
         }
         this.newComment = ''
       },
       deleteComment(id){
         if (localStorage.getItem('userData') === null) {
           alert('User not authorised.')
           return
         }
         const userData = JSON.parse(localStorage.getItem('userData'));
         console.log(id)
         console.log(userData.accessToken)
         this.$http.delete('/comment/delete', { params: { id, accessToken: userData.accessToken } }).then(()
=> {
           this.$http.get('/comment/emoteid', { params: { emote_id: this.emote.id } }).then((response) =>
this.comments = [...response.data])
         }).catch((error) => alert(error.response.data.statusText))
       }
     }
   }
```

В итоге, получился следующий результат:

Главная страница



Twitch cancer emotes album

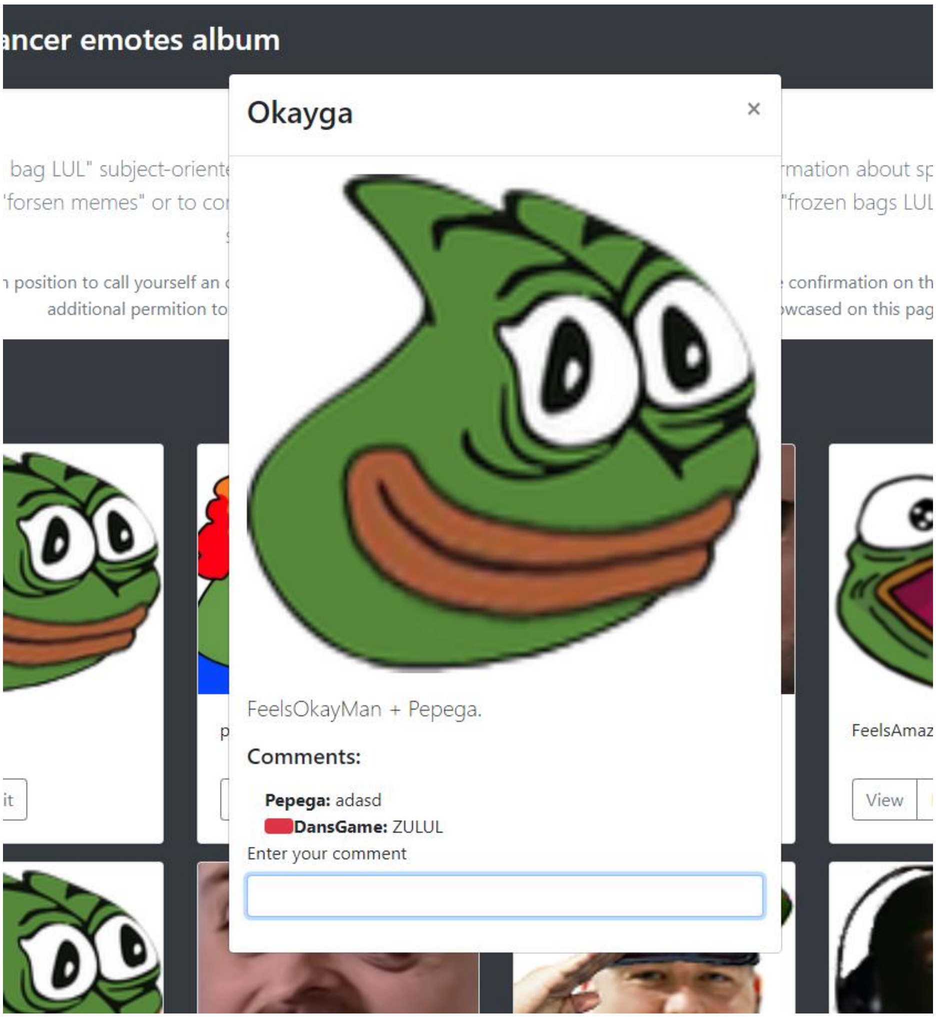
SUPA Website

This "frozen bag LUL" subject-oriented website is the suitable place for you to aquire critical information about specific twitch meta known as "forsen memes" or to communicate with other subjects which identify themselves as "frozen bags LUL". Here the wide spectre of twitch emotes are available for your exploration.

If you are in position to call yourself an official mebmer of elite group identified as "Streamsavers" (and provide confirmation on that topic), you gain additional permition to modify and complement any information related to twitch emoted showcased on this page

Twitch emotes

Просмотр эмоции и комментариев (красный прямоугольник означает возможность удаления своего комментария)



Okayga

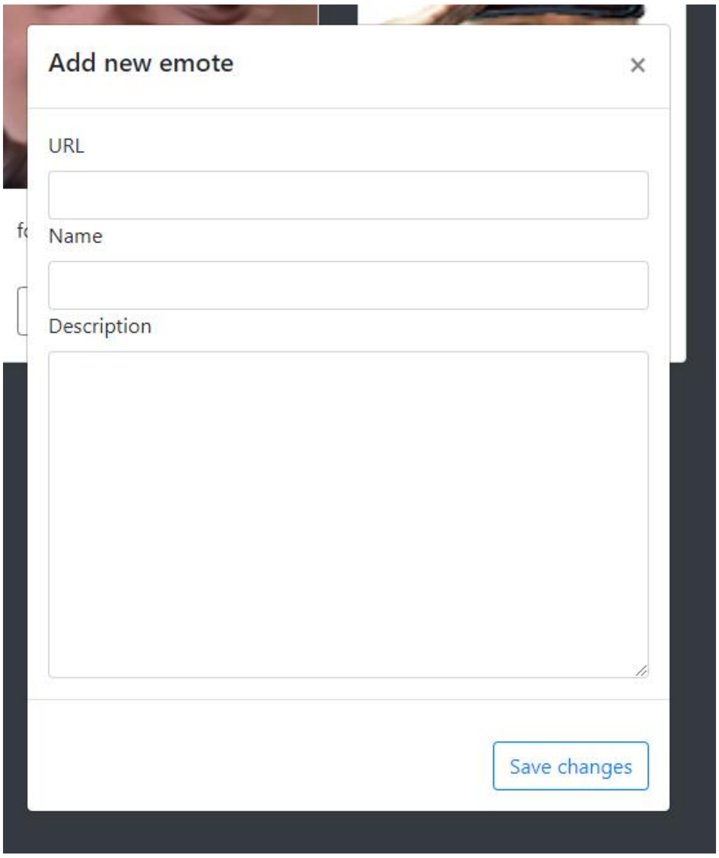
FeelsOkayMan + Pepega.

Comments:

Pepega: adasd
🟥**DansGame:** ZULUL

Enter your comment

Создание эмоции



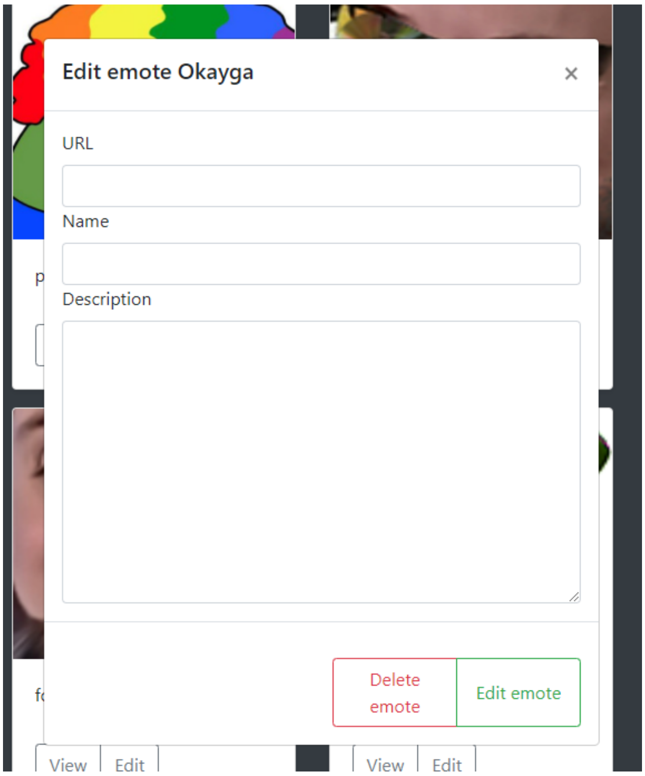
Add new emote

URL

Name

Description

Изменение эмоции



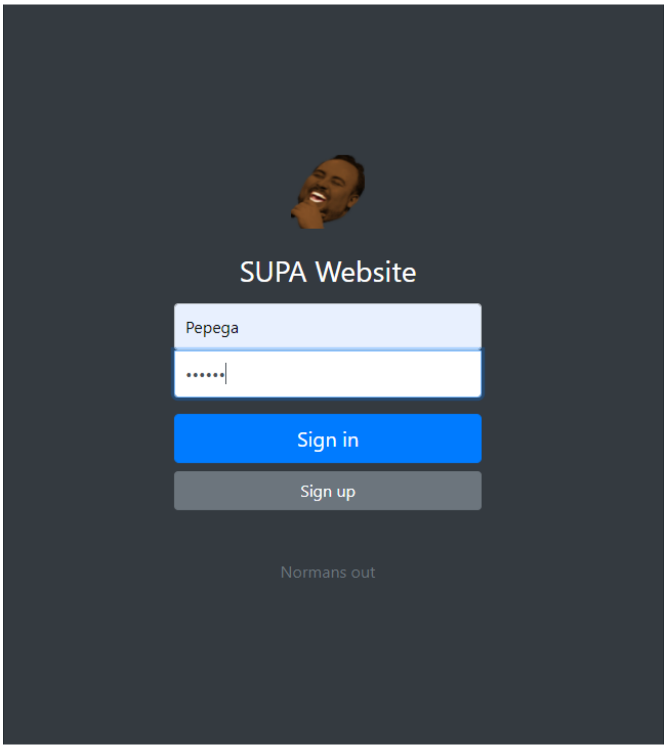
Edit emote Okayga

URL

Name

Description

Вход



SUPA Website

Pepega

••••••

Вывод

В результате выполнения данной работы были изучены основы работы протокола передачи гипертекста HTTP и языка CSS и основы разработки frontend-приложений, в частности, с использованием клиентских фреймворков (Vue.js). Также были рассмотрены основы работы docker и фреймворком Yii2 и

Изучены основы взаимодействия web-приложения с базой данных и принципы работы отправки Ajax запросов.