# Solutions-JS Apps Exam – Spotify Single Page Application

//This solution don’t work good!

//index.html

<!-- The remove work, but not good, like and listen don't work, mySongs don't work! -->

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Spotify</title>

<!-- styles -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"

integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

<link rel="stylesheet" href="./styles/site.css">

<link rel="stylesheet" href="./styles/notifications.css">

<!-- node modules scripts -->

<script src="./node\_modules/jquery/dist/jquery.min.js"></script>

<script src="./node\_modules/handlebars/dist/handlebars.min.js"></script>

<script src="./node\_modules/sammy/lib/sammy.js"></script>

<script src="./node\_modules/sammy/lib/plugins/sammy.handlebars.js"></script>

<!-- default scripts -->

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

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

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

<!-- services -->

<script src="./scripts/services/user-service.js"></script>

<script src="./scripts/services/song-service.js"></script>

<!-- handlers -->

<script src="./scripts/handlers/user-handler.js"></script>

<script src="./scripts/handlers/home-handler.js"></script>

<script src="./scripts/handlers/song-handler.js"></script>

</head>

<body>

<div id="container"></div>

<div id="notifications">

<div id="loadingBox" class="notification">

<span>Loading …</span>

</div>

<div id="infoBox" class="notification">

<span>Info</span>

</div>

<div id="errorBox" class="notification">

<span>Error</span>

</div>

</div>

</body>

</html>

//package-lock.json

{

"requires": true,

"lockfileVersion": 1,

"dependencies": {

"commander": {

"version": "2.20.0",

"resolved": "https://registry.npmjs.org/commander/-/commander-2.20.0.tgz",

"integrity": "sha512-7j2y+40w61zy6YC2iRNpUe/NwhNyoXrYpHMrSunaMG64nRnaf96zO/KMQR4OyN/UnE5KLyEBnKHd4aG3rskjpQ==",

"optional": true

},

"handlebars": {

"version": "4.1.2",

"resolved": "https://registry.npmjs.org/handlebars/-/handlebars-4.1.2.tgz",

"integrity": "sha512-nvfrjqvt9xQ8Z/w0ijewdD/vvWDTOweBUm96NTr66Wfvo1mJenBLwcYmPs3TIBP5ruzYGD7Hx/DaM9RmhroGPw==",

"requires": {

"neo-async": "^2.6.0",

"optimist": "^0.6.1",

"source-map": "^0.6.1",

"uglify-js": "^3.1.4"

}

},

"jquery": {

"version": "3.4.0",

"resolved": "https://registry.npmjs.org/jquery/-/jquery-3.4.0.tgz",

"integrity": "sha512-ggRCXln9zEqv6OqAGXFEcshF5dSBvCkzj6Gm2gzuR5fWawaX8t7cxKVkkygKODrDAzKdoYw3l/e3pm3vlT4IbQ=="

},

"minimist": {

"version": "0.0.10",

"resolved": "https://registry.npmjs.org/minimist/-/minimist-0.0.10.tgz",

"integrity": "sha1-3j+YVD2/lggr5IrRoMfNqDYwHc8="

},

"neo-async": {

"version": "2.6.0",

"resolved": "https://registry.npmjs.org/neo-async/-/neo-async-2.6.0.tgz",

"integrity": "sha512-MFh0d/Wa7vkKO3Y3LlacqAEeHK0mckVqzDieUKTT+KGxi+zIpeVsFxymkIiRpbpDziHc290Xr9A1O4Om7otoRA=="

},

"optimist": {

"version": "0.6.1",

"resolved": "https://registry.npmjs.org/optimist/-/optimist-0.6.1.tgz",

"integrity": "sha1-2j6nRob6IaGaERwybpDrFaAZZoY=",

"requires": {

"minimist": "~0.0.1",

"wordwrap": "~0.0.2"

}

},

"sammy": {

"version": "0.7.6",

"resolved": "https://registry.npmjs.org/sammy/-/sammy-0.7.6.tgz",

"integrity": "sha1-Rk/lTR+JZhzDqYgAYwVxEyHpqcE="

},

"source-map": {

"version": "0.6.1",

"resolved": "https://registry.npmjs.org/source-map/-/source-map-0.6.1.tgz",

"integrity": "sha512-UjgapumWlbMhkBgzT7Ykc5YXUT46F0iKu8SGXq0bcwP5dz/h0Plj6enJqjz1Zbq2l5WaqYnrVbwWOWMyF3F47g=="

},

"uglify-js": {

"version": "3.5.5",

"resolved": "https://registry.npmjs.org/uglify-js/-/uglify-js-3.5.5.tgz",

"integrity": "sha512-e58FqZzPwaLODQetDQKlvErZaGkh1UmzP8YwU0aG65NLourKNtwVyDG8tkIyUU0vqWzxaikSvTaxrCSscmvqvQ==",

"optional": true,

"requires": {

"commander": "~2.20.0",

"source-map": "~0.6.1"

}

},

"wordwrap": {

"version": "0.0.3",

"resolved": "https://registry.npmjs.org/wordwrap/-/wordwrap-0.0.3.tgz",

"integrity": "sha1-o9XabNXAvAAI03I0u68b7WMFkQc="

}

}

}

//package.json

{

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"handlebars": "^4.1.1",

"jquery": "^3.3.1",

"sammy": "^0.7.6"

}

}

//scripts.app.js

const handlers = {}

$(() => {

const app = Sammy('#container', function () {

this.use('Handlebars', 'hbs');

// home page routes

this.get('/index.html', handlers.getHome);

this.get('/', handlers.getHome);

this.get('#/home', handlers.getHome);

// user routes

this.get('#/register', handlers.getRegister);

this.get('#/login', handlers.getLogin);

this.post('#/register', handlers.registerUser);

this.post('#/login', handlers.loginUser);

this.get('#/logout', handlers.logoutUser);

// songs

this.get('#/allSongs', handlers.getAllSongs);

this.get('#/createSong', handlers.getCreateSong);

this.post('#/createSong', handlers.createSong);

this.get('#/like/:id', handlers.likeSong);

this.get('#/remove/:id', handlers.removeSong);

this.get('#/listen/:id', handlers.listenSong);

this.get('#/mySongs', handlers.getMySongs);

});

app.run('#/home');

});

//scripts.kinvey.js

const kinvey = (() => {

const BASE\_URL = 'https://baas.kinvey.com/';

const APP\_KEY = 'kid\_H1y9kL-\_N';

const APP\_SECRET = '270909213f424f249f5d6da36f8c31f3';

function makeAuth(auth) {

if (auth == 'basic') {

return {

'Authorization': `Basic ${btoa(APP\_KEY + ':' + APP\_SECRET)}`

}

} else {

return {

'Authorization': `Kinvey ${sessionStorage.getItem('authtoken')}`

}

}

}

function makeRequest(method, collection, endpoint, auth) {

return {

url: BASE\_URL + collection + '/' + APP\_KEY + '/' + endpoint,

method,

headers: makeAuth(auth)

}

}

function post(collection, endpoint, auth, data) {

let req = makeRequest('POST', collection, endpoint, auth);

req.data = data;

return $.ajax(req);

}

function get(collection, endpoint, auth, query) {

if (query) {

endpoint += query;

}

return $.ajax(makeRequest('GET', collection, endpoint, auth));

}

function update(collection, endpoint, auth, data) {

let req = makeRequest('PUT', collection, endpoint, auth);

req.data = data;

return $.ajax(req);

}

function remove(collection, endpoint, auth) {

return $.ajax(makeRequest('DELETE', collection, endpoint, auth));

}

return {

get,

post,

update,

remove

}

})();

//scripts.notify.js

let notify = (() => {

$(document).on({

ajaxStart: () => $('#loadingBox').show(),

ajaxStop: () => $('#loadingBox').fadeOut()

})

function showInfo (message) {

let infoBox = $('#infoBox')

infoBox.find('span').text(message)

infoBox.fadeIn()

setTimeout(() => infoBox.fadeOut(), 3000)

}

function showError(message) {

let errorBox = $('#errorBox')

errorBox.find('span').text(message)

errorBox.fadeIn()

setTimeout(() => errorBox.fadeOut(), 3000)

}

function handleError (reason) {

showError(reason.responseJSON.description)

}

return {

showInfo,

showError,

handleError

}

})();

//scripts.handlers.home-handler.js

handlers.getHome = function (context) {

context.isAuth = userService.isAuth();

context.username = sessionStorage.getItem('username');

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs'

}).then(function () {

this.partial('./templates/home.hbs');

}).catch(function (error) {

notify.handleError(error);

});

}

//scripts.handlers.song-handler.js

handlers.getAllSongs = function(context) {

context.isAuth = userService.isAuth();

context.username = sessionStorage.getItem('username');

songService.getAllSongs().then(function(response) {

let userId = sessionStorage.getItem('id');

let songs = response;

songs.forEach(song => song.isCreator = song.\_acl.creator === userId);

songs = songs.sort((a, b) => a.isCreator - b.isCreator);

context.songs = songs;

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

song: './templates/song/song.hbs'

}).then(function() {

this.partial('./templates/song/allSongs.hbs');

}).catch(function(err) {

notify.handleError(err);

});

}).catch(function(error) {

notify.handleError(error);

});

};

handlers.getCreateSong = function(context) {

context.isAuth = userService.isAuth();

context.userService = sessionStorage.getItem('username');

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs'

}).then(function() {

this.partial('./templates/song/createSong.hbs');

}).catch(function(error) {

notify.handleError(error);

});

};

handlers.createSong = function(context) {

let data = {...context.params, likeCounter: 0, listenCounter: 0};

if (data.title.length < 6) {

notify.showError('The title should be at least 6 characters long!');

return;

}

if (data.artist.length < 3){

notify.showError('The artist should be at least 3 characters long!');

return;

}

if (!data.imageURL.startsWith('http://') && !data.imageURL.startsWith('https://')){

notify.showError('The image URL should start with "http://" or "https//"');

return;

}

songService.createSong(data).then(function(response) {

notify.showInfo('Song created successfully!');

context.redirect('#/allSongs');

}).catch(function(error) {

notify.handleError(error);

});

};

handlers.likeSong = function(context) {

let id = context.params.id;

songService.getASong(id).then(function(response) {

let song = response;

let newLikeCounter = Number(song.likeCounter) + 1;

song.likeCounter = newLikeCounter;

songService.likeSong(id, song).then(function() {

notify.showInfo('Liked');

context.redirect('#/allSongs');

}).catch(function(err) {

notify.handleError(err);

});

}).catch(function(error) {

notify.handleError(error);

});

};

handlers.listenSong = function(context) {

let id = context.params.id;

songService.getASong(id).then(function(response) {

let song = response;

let newListenCounter = Number(song.listenCounter) + 1;

song.listenCounter = newListenCounter;

songService.listenSong(id, song).then(function() {

notify.showInfo(`You just litened ${song.title}`);

context.redirect('#/allSongs');

}).catch(function(err) {

notify.handleError(err);

});

}).catch(function(error) {

notify.handleError(error);

})

};

handlers.removeSong = function(context) {

let id = context.params.id;

songService.removeSong(id).then(function() {

notify.showInfo('Song removed successfully!');

context.redirect('#/allSongs');

}).catch(function(error) {

notify.handleError(error);

});

};

handlers.getMySongs = function(context) {

context.isAuth = userService.isAuth();

context.username = sessionStorage.getItem('username');

songService.getAllMySongs().then(function(response) {

let userId = sessionStorage.getItem('id');

let songs = response;

songs.forEach(song => song.isCreator = song.\_acl.creator === userId);

context.songs = songs;

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs',

song: './templates/song/song.hbs'

}).then(function() {

this.partial('./templates/song/mySongs.hbs');

}).catch(function(err) {

notify.handleError(err);

});

}).catch(function(error) {

notify.handleError(error);

});

};

//scripts.handlers.user-handler.js

handlers.getRegister = function (context) {

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs'

}).then(function () {

this.partial('./templates/register.hbs');

}).catch(function (error) {

notify.handleError(error);

});

}

handlers.getLogin = function (context) {

context.loadPartials({

header: './templates/common/header.hbs',

footer: './templates/common/footer.hbs'

}).then(function () {

this.partial('./templates/login.hbs');

}).catch(function (error) {

notify.handleError(error);

});

}

handlers.registerUser = function (context) {

let username = context.params.username;

let password = context.params.password;

if (username.length < 3) {

notify.showError('Username should be at least 3 character long.');

return;

}

if (password.length < 6) {

notify.showError('Password should be at least 6 character long.');

return;

}

userService.register(username, password).then((response) => {

userService.saveSession(response);

notify.showInfo('User registration successful.');

context.redirect('#/home');

}).catch(function (error) {

notify.handleError(error);

});

}

handlers.loginUser = function (context) {

let username = context.params.username;

let password = context.params.password;

userService.login(username, password).then((response) => {

userService.saveSession(response);

notify.showInfo('Login successful.');

context.redirect('#/home');

}).catch(function (error) {

notify.handleError(error);

});

}

handlers.logoutUser = function (context) {

userService.logout().then(() => {

sessionStorage.clear();

notify.showInfo('Logout successful.');

context.redirect('#/home');

})

}

//scripts.services.song-service.js

const songService = (() => {

function getAllSongs(){

return kinvey.get('appdata', 'songs', 'kinvey', `?query={}&sort={"likeCounter": -1, "listenCounter": -1}`);

}

function createSong(data) {

return kinvey.post('appdata', 'songs', 'kinvey', data)

}

function getASong(id) {

kinvey.get('appdata', `songs/${id}`, 'kinvey');

}

function likeSong(id, song) {

kinvey.update('appdata', `songs/${id}`, 'kinvey', song);

}

function listenSong(id, song) {

kinvey.update('appdata', `songs/${id}`, 'kinvey', song);

}

function removeSong(id) {

kinvey.remove('appdata', `songs/${id}`, 'kinvey');

}

function getAllMySongs() {

return kinvey.get('appdata', 'songs', 'kinvey', `?query={"\_acl.creator":"${sessionStorage.getItem('id')}"}&sort={"likeCounter": -1, "listenCounter": -1}`);

}

return {

getAllSongs,

createSong,

getASong,

likeSong,

listenSong,

removeSong,

getAllMySongs

};

})();

//scripts.services.user-service.js

const userService = (() => {

function isAuth() {

return sessionStorage.getItem('authtoken') !== null;

}

function saveSession(res) {

sessionStorage.setItem('username', res.username);

sessionStorage.setItem('authtoken', res.\_kmd.authtoken);

sessionStorage.setItem('id', res.\_id);

}

function register(username, password) {

return kinvey.post('user', '', 'basic', {

username,

password

});

}

function login(username, password) {

return kinvey.post('user', 'login', 'basic', {

username,

password

});

}

function logout() {

return kinvey.post('user', '\_logout', 'kinvey');

}

return {

register,

login,

logout,

saveSession,

isAuth

}

})();

//styles.notifications.css

#notifications {

float: right;

position: absolute;

top: 50px;

right: 0;

text-align: center;

margin: 1em;

}

.notification {

font-size: 1.5em;

color: #ffffff;

display: none;

text-align: right;

}

.notification>span {

display: inline-block;

padding: 0.5em 2em 0.5em 2em;

border-radius: 10px;

}

#loadingBox>span {

background: #7CB3E9;

}

#infoBox>span {

background: #393;

}

#errorBox>span {

background: #F50;

}

//styles.site.css

body {

background: #333333;

}

.background-spotify {

width: 100%;

}

.background-image {

height: 100vh;

}

.new-song-btn {

background: white;

color: #333333;

text-transform: uppercase;

font-weight: 500;

margin-bottom: 34px;

border: 2px solid white;

}

.new-song-btn:hover {

cursor: pointer;

color: white;

background: #333333;

border: 2px solid white;

text-decoration: none !important;

}

.new-song-btn:hover a {

text-decoration: none !important;

}

h1 {

padding: 15px;

color: white;

}

.song {

width: 23%;

display: inline-block;

margin: 20px 5px;

background-color: white;

color: black;

text-align: center;

border: 3px solid white;

padding: 15px;

height: 387px;

vertical-align: top;

}

.song-container {

width: 74%;

margin: 20px auto auto;

padding: 50px 0;

min-height: 800px;

}

.btn-block {

display: block;

width: 32%;

margin-left: 34%;

font-weight: bold;

}

.cover {

height: 154px;

width: 90%;

border: 2px solid #333333;

padding: 0;

margin: 10px;

}

p {

font-style: italic;

margin-top: 0;

margin-bottom: 0;

font-weight: bold;

color: #333333;

}

h1, h4 {

text-align: center;

}

footer {

background-color: #f8f9fa !important;

}

form {

display: inline-block;

text-align: center;

margin-left: 40%;

}

a {

color:white;

text-decoration:none

}

.white-labels {

color: white;

text-align: left !important;

font-size: 1.3rem;

}

.btn-success {

background: #4B8B13;

}

.btn-danger {

background: #FF0100;

}

.add-link {

font-size: 1.5rem;

font-style: italic;

font-weight: 400;

text-decoration: underline;

}

form {

text-align: left;

font-size: 2rem;

margin-left: 33%;

width: 36%;

}

//templates.home.hbs

{{> header}}

<section id="homeView">

<img class="m-auto background-image" width="100%" src="http://gomera24.com/wp-content/uploads/2017/06/566771.jpg">

</section>

{{> footer}}

//templates.login.hbs

{{> header}}

<section id="loginView">

<div class="background-spotify">

<div class="song-container">

<h1>Login</h1>

<form action="#/login" method="POST">

<div class="form-group">

<label for="username" class="white-labels">Username</label>

<input id="username" type="text" name="username" class="form-control" placeholder="Enter username">

</div>

<div class="form-group">

<label for="password" class="white-labels">Password</label>

<input id="password" type="password" name="password" class="form-control" placeholder="Password">

</div>

<button type="submit" class="btn btn-primary">Login</button>

</form>

<h4 class="mt-3 text-white">No account yet? <a href="#/register" class="add-link">Register</a></h4>

</div>

</div>

</section>

{{> footer}}

//templates.register.hbs

{{> header}}

<section id="registerView">

<div class="background-spotify">

<div class="song-container">

<h1>Register</h1>

<form action="#/register" method="POST">

<div class="form-group">

<label for="username" class="white-labels">Username</label>

<input type="text" name="username" class="form-control" placeholder="Enter username">

</div>

<div class="form-group">

<label for="password" class="white-labels">Password</label>

<input type="password" name="password" class="form-control" placeholder="Password">

</div>

<button type="submit" class="btn btn-primary">Register</button>

</form>

<h4 class="mt-3 text-white">Already have an account? <a href="#/login" class="add-link">Login</a></h4>

</div>

</div>

</section>

{{> footer}}

//templates.common.footer.hbs

<footer class="card-footer text-muted d-flex justify-content-center">

&copy; SoftUni JS Core Apps Exams

</footer>

//templates.common.header.hbs

<div id="container">

<nav class="navbar navbar-expand-lg navbar-light bg-light fixed-top">

<div class="collapse navbar-collapse" id="navbarText">

{{#if isAuth}}

<ul class="navbar-nav mr-auto">

<li class="nav-item active">

<a class="nav-link" href="#/home">Home <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="#/allSongs">All Songs</a>

</li>

<li class="nav-item">

<a class="nav-link " href="#/mySongs">My Songs</a>

</li>

</ul>

<ul class="navbar-nav justify-content-end">

<li class="nav-item">

<a class="nav-link" href="#">Welcome, {{username}}!</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#/logout">Logout</a>

</li>

</ul>

{{else}}

<ul class="navbar-nav mr-auto">

</ul>

<ul class="navbar-nav justify-content-end">

<li class="nav-item">

<a class="nav-link" href="#/login">Login</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#/register">Register</a>

</li>

</ul>

</div>

{{/if}}

</nav>

//templates.song.allSongs.hbs

{{> header}}

<section id="allSongsView">

<div class="background-spotify">

<div class="song-container">

<h1>All Songs</h1>

<a href="#/createSong">

<button type="button" class="btn-lg btn-block new-song-btn">Add a new song</button>

</a>

{{#each songs}}

{{> song}}

{{/each}}

</div>

</div>

</section>

{{> footer}}

//templates.song.createSong.hbs

{{> header}}

<section id="createSongView">

<div class="background-spotify">

<div class="song-container">

<h1>Create new song</h1>

<form action="#/createSong" method="POST">

<div class="form-group">

<label for="title" class="white-labels">Title</label>

<input id="title" type="text" name="title" class="form-control" placeholder="Title">

</div>

<div class="form-group">

<label for="artist" class="white-labels">Artist</label>

<input id="artist" type="text" name="artist" class="form-control" placeholder="Artist">

</div>

<div class="form-group">

<label for="imageURL" class="white-labels">imageURL</label>

<input id="imageURL" type="text" name="imageURL" class="form-control" placeholder="imageURL">

</div>

<button type="submit" class="btn btn-primary">Create</button>

</form>

</div>

</div>

</section>

{{> footer}}

//templates.song.mySongs.hbs

{{> header}}

<section id="mySongsView">

<div class="background-spotify">

<div class="song-container">

<h1>My Songs</h1>

{{#each songs}}

{{song}}

{{/each}}

</div>

</div>

</section>

{{> footer}}

//templates.song.song.hbs

<div class="song" id="{{id}}">

<h5>Title: {{title}}</h5>

<h5>Artist: {{artist}}</h5>

<img class="cover" src="{{imageURL}}" />

{{#if isCreator}}

<p>Likes: {{likeCounter}}; Listened {{listenCounter}} times</p>

<a href="#/remove/{{\_id}}"><button type="button" class="btn btn-danger mt-4">Remove</button></a>

<a href="#/listen/{{\_id}}"><button type="button" class="btn btn-success mt-4">Listen</button></a>

{{else}}

<p>Likes: {{likeCounter}}</p>

<a href="#/like/{{\_id}}"><button type="button" class="btn btn-primary mt-4">Like</button></a>

{{/if}}

</div>

//full solution from Aleksander Ovardov

## ****//index.html****

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Skeleton</title> |
|  | <!-- styles --> |
|  | <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" |
|  | integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous"> |
|  | <link rel="stylesheet" href="./styles/site.css"> |
|  | <link rel="stylesheet" href="./styles/notifications.css"> |
|  |  |
|  | <!-- node modules scripts --> |
|  | <script src="./node\_modules/jquery/dist/jquery.min.js"></script> |
|  | <script src="./node\_modules/handlebars/dist/handlebars.min.js"></script> |
|  | <script src="./node\_modules/sammy/lib/sammy.js"></script> |
|  | <script src="./node\_modules/sammy/lib/plugins/sammy.handlebars.js"></script> |
|  |  |
|  | <!-- default scripts --> |
|  | <script src="./scripts/kinvey.js"></script> |
|  | <script src="./scripts/app.js"></script> |
|  | <script src="./scripts/notify.js"></script> |
|  |  |
|  | <!-- services --> |
|  | <script src="./scripts/services/user-service.js"></script> |
|  | <script src="./scripts/services/song-service.js"></script> |
|  |  |
|  | <!-- handlers --> |
|  | <script src="./scripts/handlers/user-handler.js"></script> |
|  | <script src="./scripts/handlers/home-handler.js"></script> |
|  | <script src="./scripts/handlers/song-handler.js"></script> |
|  |  |
|  | </head> |
|  |  |
|  | <body> |
|  |  |
|  | <div id="container"></div> |
|  |  |
|  | <div id="notifications"> |
|  | <div id="loadingBox" class="notification"> |
|  | <span>Loading …</span> |
|  | </div> |
|  | <div id="infoBox" class="notification"> |
|  | <span>Info</span> |
|  | </div> |
|  | <div id="errorBox" class="notification"> |
|  | <span>Error</span> |
|  | </div> |
|  | </div> |
|  |  |
|  | </body> |
|  |  |
|  | </html> |

## //****package-lock.json****

|  |
| --- |
| { |
|  | "name": "apps-skeleton", |
|  | "version": "1.0.0", |
|  | "lockfileVersion": 1, |
|  | "requires": true, |
|  | "dependencies": { |
|  | "commander": { |
|  | "version": "2.19.0", |
|  | "resolved": "https://registry.npmjs.org/commander/-/commander-2.19.0.tgz", |
|  | "integrity": "sha512-6tvAOO+D6OENvRAh524Dh9jcfKTYDQAqvqezbCW82xj5X0pSrcpxtvRKHLG0yBY6SD7PSDrJaj+0AiOcKVd1Xg==", |
|  | "optional": true |
|  | }, |
|  | "handlebars": { |
|  | "version": "4.1.1", |
|  | "resolved": "https://registry.npmjs.org/handlebars/-/handlebars-4.1.1.tgz", |
|  | "integrity": "sha512-3Zhi6C0euYZL5sM0Zcy7lInLXKQ+YLcF/olbN010mzGQ4XVm50JeyBnMqofHh696GrciGruC7kCcApPDJvVgwA==", |
|  | "requires": { |
|  | "neo-async": "^2.6.0", |
|  | "optimist": "^0.6.1", |
|  | "source-map": "^0.6.1", |
|  | "uglify-js": "^3.1.4" |
|  | } |
|  | }, |
|  | "jquery": { |
|  | "version": "3.3.1", |
|  | "resolved": "https://registry.npmjs.org/jquery/-/jquery-3.3.1.tgz", |
|  | "integrity": "sha512-Ubldcmxp5np52/ENotGxlLe6aGMvmF4R8S6tZjsP6Knsaxd/xp3Zrh50cG93lR6nPXyUFwzN3ZSOQI0wRJNdGg==" |
|  | }, |
|  | "minimist": { |
|  | "version": "0.0.10", |
|  | "resolved": "https://registry.npmjs.org/minimist/-/minimist-0.0.10.tgz", |
|  | "integrity": "sha1-3j+YVD2/lggr5IrRoMfNqDYwHc8=" |
|  | }, |
|  | "neo-async": { |
|  | "version": "2.6.0", |
|  | "resolved": "https://registry.npmjs.org/neo-async/-/neo-async-2.6.0.tgz", |
|  | "integrity": "sha512-MFh0d/Wa7vkKO3Y3LlacqAEeHK0mckVqzDieUKTT+KGxi+zIpeVsFxymkIiRpbpDziHc290Xr9A1O4Om7otoRA==" |
|  | }, |
|  | "optimist": { |
|  | "version": "0.6.1", |
|  | "resolved": "https://registry.npmjs.org/optimist/-/optimist-0.6.1.tgz", |
|  | "integrity": "sha1-2j6nRob6IaGaERwybpDrFaAZZoY=", |
|  | "requires": { |
|  | "minimist": "~0.0.1", |
|  | "wordwrap": "~0.0.2" |
|  | } |
|  | }, |
|  | "sammy": { |
|  | "version": "0.7.6", |
|  | "resolved": "https://registry.npmjs.org/sammy/-/sammy-0.7.6.tgz", |
|  | "integrity": "sha1-Rk/lTR+JZhzDqYgAYwVxEyHpqcE=" |
|  | }, |
|  | "source-map": { |
|  | "version": "0.6.1", |
|  | "resolved": "https://registry.npmjs.org/source-map/-/source-map-0.6.1.tgz", |
|  | "integrity": "sha512-UjgapumWlbMhkBgzT7Ykc5YXUT46F0iKu8SGXq0bcwP5dz/h0Plj6enJqjz1Zbq2l5WaqYnrVbwWOWMyF3F47g==" |
|  | }, |
|  | "uglify-js": { |
|  | "version": "3.5.3", |
|  | "resolved": "https://registry.npmjs.org/uglify-js/-/uglify-js-3.5.3.tgz", |
|  | "integrity": "sha512-rIQPT2UMDnk4jRX+w4WO84/pebU2jiLsjgIyrCktYgSvx28enOE3iYQMr+BD1rHiitWnDmpu0cY/LfIEpKcjcw==", |
|  | "optional": true, |
|  | "requires": { |
|  | "commander": "~2.19.0", |
|  | "source-map": "~0.6.1" |
|  | } |
|  | }, |
|  | "wordwrap": { |
|  | "version": "0.0.3", |
|  | "resolved": "https://registry.npmjs.org/wordwrap/-/wordwrap-0.0.3.tgz", |
|  | "integrity": "sha1-o9XabNXAvAAI03I0u68b7WMFkQc=" |
|  | } |
|  | } |
|  | } |

## //****package.json****

|  |
| --- |
| { |
|  | "name": "apps-skeleton", |
|  | "version": "1.0.0", |
|  | "description": "", |
|  | "main": "index.js", |
|  | "scripts": { |
|  | "test": "echo \"Error: no test specified\" && exit 1" |
|  | }, |
|  | "keywords": [], |
|  | "author": "", |
|  | "license": "ISC", |
|  | "dependencies": { |
|  | "handlebars": "^4.1.1", |
|  | "jquery": "^3.3.1", |
|  | "sammy": "^0.7.6" |
|  | } |
|  | } |

//scripts.app.js

|  |
| --- |
| const handlers = {}; |
|  |  |
|  | $(() => { |
|  | const app = Sammy('#container', function () { |
|  | this.use('Handlebars', 'hbs'); |
|  |  |
|  | // home page routes |
|  | this.get('/index.html', handlers.getHome); |
|  | this.get('/', handlers.getHome); |
|  | this.get('#/home', handlers.getHome); |
|  |  |
|  | // user routes |
|  | this.get('#/register', handlers.getRegister); |
|  | this.get('#/login', handlers.getLogin); |
|  |  |
|  | this.post('#/register', handlers.registerUser); |
|  | this.post('#/login', handlers.loginUser); |
|  |  |
|  | this.get('#/logout', handlers.logoutUser); |
|  |  |
|  | // Songs |
|  | this.get('#/allSongs', handlers.getAllSongs); |
|  | this.get('#/createSong', handlers.getCreateSong); |
|  | this.get('#/mySongs', handlers.getMySongs); |
|  |  |
|  | this.post('#/createSong', handlers.createSong); |
|  |  |
|  | this.get('#/remove/:id', handlers.removeSong); |
|  | this.get('#/like/:id', handlers.likeSong); |
|  | this.get('#/listen/:id', handlers.listenSong); |
|  |  |
|  | }); |
|  |  |
|  | app.run('#/home'); |
|  | }); |

//scripts.kinvey.js

|  |
| --- |
| const kinvey = (() => { |
|  | const BASE\_URL = 'https://baas.kinvey.com/'; |
|  | const APP\_KEY = 'kid\_H1vHdJ9KE'; |
|  | const APP\_SECRET = 'c706aa9e3f6c495cb48d176b1cb1e88e'; |
|  |  |
|  | function makeAuth(auth) { |
|  | if (auth == 'basic') { |
|  | return { |
|  | 'Authorization': `Basic ${btoa(APP\_KEY + ':' + APP\_SECRET)}` |
|  | } |
|  | } else { |
|  | return { |
|  | 'Authorization': `Kinvey ${sessionStorage.getItem('authtoken')}` |
|  | } |
|  | } |
|  | } |
|  |  |
|  | function makeRequest(method, collection, endpoint, auth) { |
|  | return { |
|  | url: BASE\_URL + collection + '/' + APP\_KEY + '/' + endpoint, |
|  | method, |
|  | headers: makeAuth(auth) |
|  | } |
|  | } |
|  |  |
|  | function post(collection, endpoint, auth, data) { |
|  | let req = makeRequest('POST', collection, endpoint, auth); |
|  | req.data = data; |
|  | return $.ajax(req); |
|  | } |
|  |  |
|  | function get(collection, endpoint, auth, query) { |
|  | if(query){ |
|  | endpoint += query |
|  | } |
|  |  |
|  | return $.ajax(makeRequest('GET', collection, endpoint, auth)); |
|  | } |
|  |  |
|  | function update(collection, endpoint, auth, data) { |
|  | let req = makeRequest('PUT', collection, endpoint, auth); |
|  | req.data = data; |
|  | return $.ajax(req); |
|  | } |
|  |  |
|  | function remove(collection, endpoint, auth) { |
|  | return $.ajax(makeRequest('DELETE', collection, endpoint, auth)); |
|  | } |
|  |  |
|  | return { |
|  | get, |
|  | post, |
|  | update, |
|  | remove |
|  | } |
|  | })(); |

## **//scripts.**notify.js****

|  |
| --- |
| let notify = (() => { |
|  | $(document).on({ |
|  | ajaxStart: () => $('#loadingBox').show(), |
|  | ajaxStop: () => $('#loadingBox').fadeOut() |
|  | }); |
|  |  |
|  | function showInfo (message) { |
|  | let infoBox = $('#infoBox'); |
|  | infoBox.find('span').text(message); |
|  | infoBox.fadeIn(); |
|  | setTimeout(() => infoBox.fadeOut(), 3000) |
|  | } |
|  |  |
|  | function showError(message) { |
|  | let errorBox = $('#errorBox'); |
|  | errorBox.find('span').text(message); |
|  | errorBox.fadeIn(); |
|  | setTimeout(() => errorBox.fadeOut(), 3000) |
|  | } |
|  |  |
|  | function handleError (reason) { |
|  | showError(reason.responseJSON.description) |
|  | } |
|  |  |
|  | return { |
|  | showInfo, |
|  | showError, |
|  | handleError |
|  | } |
|  | })(); |

//scripts.handlers.home-handler.js

|  |
| --- |
| handlers.getHome = function (ctx) { |
|  | ctx.isAuth = userService.isAuth(); |
|  | ctx.username = sessionStorage.getItem('username'); |
|  |  |
|  | ctx.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs' |
|  | }) |
|  | .then(function () { |
|  | this.partial('./templates/home.hbs'); |
|  | }) |
|  | .catch(function (err) { |
|  | notify.handleError(err); |
|  | }); |
|  | }; |

//scripts.handlers.song-handler.js

|  |
| --- |
| handlers.getAllSongs = function (context) { |
|  | context.isAuth = userService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | songService.getAllSongs() |
|  | .then(function (res) { |
|  | let userId = sessionStorage.getItem('id'); |
|  | let songs = res; |
|  |  |
|  | songs.forEach((song) => song.isCreator = song.\_acl.creator === userId); |
|  | songs = songs.sort((a, b) => a.isCreator - b.isCreator); |
|  |  |
|  | context.songs = songs; |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | song: './templates/song/song.hbs' |
|  | }) |
|  | .then(function () { |
|  | this.partial('./templates/song/allSongs.hbs'); |
|  | }) |
|  | .catch(function (error) { |
|  | notify.handleError(error); |
|  | }) |
|  | }); |
|  |  |
|  | }; |
|  |  |
|  | handlers.getCreateSong = function (context) { |
|  | context.isAuth = userService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs' |
|  | }) |
|  | .then(function () { |
|  | this.partial('./templates/song/createSong.hbs'); |
|  | }) |
|  | .catch(function (error) { |
|  | notify.handleError(error); |
|  | }) |
|  | }; |
|  |  |
|  | handlers.getMySongs = function (context) { |
|  | context.isAuth = userService.isAuth(); |
|  | context.username = sessionStorage.getItem('username'); |
|  |  |
|  | songService.getAllMySongs() |
|  | .then(function (res) { |
|  | let userId = sessionStorage.getItem('id'); |
|  | let songs = res; |
|  |  |
|  | songs.forEach((song) => song.isCreator = song.\_acl.creator === userId); |
|  |  |
|  | context.songs = songs; |
|  |  |
|  | context.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs', |
|  | song: './templates/song/song.hbs' |
|  | }) |
|  | .then(function () { |
|  | this.partial('./templates/song/mySongs.hbs'); |
|  | }) |
|  | .catch(function (error) { |
|  | notify.handleError(error); |
|  | }) |
|  | }); |
|  | }; |
|  |  |
|  | handlers.createSong = function (context) { |
|  | let data = {...context.params, likeCounter: 0, listenCounter: 0}; |
|  |  |
|  | if (data.title.length < 6) { |
|  | notify.showError('The title should be at least 6 characters long!') |
|  | } else if (data.artist < 3) { |
|  | notify.showError('The artist should be at least 3 characters long!') |
|  | } else if (!data.imageURL.startsWith('https://')) { |
|  | notify.showError('The image should start with "http://" or "https://"') |
|  | } else { |
|  |  |
|  | songService.createSong(data) |
|  | .then(function (res) { |
|  |  |
|  | notify.showInfo('Song created successfully.'); |
|  | context.redirect('#/allSongs'); |
|  | }) |
|  | .catch(function (error) { |
|  | notify.handleError(error); |
|  | }); |
|  | } |
|  | }; |
|  |  |
|  | handlers.removeSong = function (context) { |
|  | let id = context.params.id; |
|  |  |
|  | songService.removeSong(id) |
|  | .then(function () { |
|  | notify.showInfo('Song removed successfully!'); |
|  | context.redirect('#/allSongs'); |
|  | }) |
|  | .catch(function (error) { |
|  | notify.handleError(error); |
|  | }) |
|  |  |
|  | }; |
|  |  |
|  | handlers.likeSong = function (context) { |
|  | let id = context.params.id; |
|  |  |
|  | songService.getASong(id) |
|  | .then(function (res) { |
|  | let song = res; |
|  |  |
|  | let newLikes = Number(song.likeCounter) + 1; |
|  | song.likeCounter = newLikes; |
|  |  |
|  | songService.likeSong(id, song) |
|  | .then(function () { |
|  | notify.showInfo('Liked'); |
|  | context.redirect('#/allSongs'); |
|  | }) |
|  | .catch(function (error) { |
|  | notify.showError(error); |
|  | }) |
|  | }); |
|  | }; |
|  |  |
|  | handlers.listenSong = function (context) { |
|  | let id = context.params.id; |
|  |  |
|  | songService.getASong(id) |
|  | .then(function (res) { |
|  | let song = res; |
|  |  |
|  | let newListen = Number(song.listenCounter) + 1; |
|  | song.listenCounter = newListen; |
|  |  |
|  | songService.likeSong(id, song) |
|  | .then(function () { |
|  |  |
|  | notify.showInfo(`You just listened ${song.title}`); |
|  | context.redirect('#/allSongs'); |
|  | }) |
|  | .catch(function (error) { |
|  | notify.showError(error); |
|  | }) |
|  | }); |
|  | }; |

//scripts.handlers.user-handler.js

|  |
| --- |
| handlers.getRegister = function (ctx) { |
|  | ctx.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs' |
|  | }) |
|  | .then(function () { |
|  | this.partial('./templates/register.hbs'); |
|  | }) |
|  | .catch(function (err) { |
|  | notify.handleError(err) |
|  | }); |
|  | }; |
|  |  |
|  | handlers.getLogin = function (ctx) { |
|  | ctx.loadPartials({ |
|  | header: './templates/common/header.hbs', |
|  | footer: './templates/common/footer.hbs' |
|  | }) |
|  | .then(function () { |
|  | this.partial('./templates/login.hbs'); |
|  | }) |
|  | .catch(function (err) { |
|  | notify.handleError(err) |
|  | }); |
|  | }; |
|  |  |
|  | handlers.registerUser = function (ctx) { |
|  | let username = ctx.params.username; |
|  | let password = ctx.params.password; |
|  |  |
|  | if(username.length < 3){ |
|  | notify.showError('Username should be at least 3 character long.') |
|  | } else if(password.length < 6){ |
|  | notify.showError('Password should be at least 6 character long.') |
|  | } else{ |
|  | userService.register(username, password) |
|  | .then((res) => { |
|  | userService.saveSession(res); |
|  | notify.showInfo('User registration successful.'); |
|  |  |
|  | ctx.redirect('#/home'); |
|  | }) |
|  | .catch(function (err) { |
|  | notify.handleError(err) |
|  | }); |
|  | } |
|  | }; |
|  |  |
|  | handlers.logoutUser = function (ctx) { |
|  | userService.logout() |
|  | .then(() => { |
|  | sessionStorage.clear(); |
|  | notify.showInfo('Logout successful.'); |
|  | ctx.redirect('#/login'); |
|  | }) |
|  | }; |
|  |  |
|  | handlers.loginUser = function (ctx) { |
|  | let username = ctx.params.username; |
|  | let password = ctx.params.password; |
|  |  |
|  | userService.login(username, password) |
|  | .then((res) => { |
|  |  |
|  | userService.saveSession(res); |
|  | notify.showInfo('Login successful.'); |
|  |  |
|  | ctx.redirect('#/home'); |
|  |  |
|  | }) |
|  | .catch(function (err) { |
|  | notify.handleError(err) |
|  | }); |
|  | }; |

//scripts.services.song-service.js

|  |
| --- |
| const songService = (() => { |
|  |  |
|  | function createSong(data) { |
|  | return kinvey.post('appdata', 'songs', 'kinvey', data); |
|  | } |
|  |  |
|  | function getAllSongs() { |
|  | return kinvey.get('appdata', 'songs', 'kinvey', `?query={}&sort={"likeCounter": -1, "listenCounter": -1}`); |
|  | } |
|  |  |
|  | function getAllMySongs() { |
|  | return kinvey.get('appdata', 'songs', 'kinvey', `?query={"\_acl.creator":"${sessionStorage.getItem('id')}"}&sort={"likeCounter": -1, "listenCounter": -1}`); |
|  | } |
|  |  |
|  | function removeSong(id) { |
|  | return kinvey.remove('appdata', `songs/${id}`, 'kinvey'); |
|  | } |
|  |  |
|  | function likeSong(id, song) { |
|  | return kinvey.update('appdata', `songs/${id}`, 'kinvey', song); |
|  | } |
|  |  |
|  | function listenSong(id, song) { |
|  | return kinvey.update('appdata', `songs/${id}`, 'kinvey', song); |
|  | } |
|  |  |
|  | function getASong(id) { |
|  | return kinvey.get('appdata', `songs/${id}`, 'kinvey'); |
|  | } |
|  |  |
|  | return { |
|  | createSong, |
|  | getAllSongs, |
|  | getAllMySongs, |
|  | removeSong, |
|  | likeSong, |
|  | listenSong, |
|  | getASong |
|  | } |
|  |  |
|  | })(); |

//scripts.services.user-service.js

|  |
| --- |
| const userService = (() => { |
|  | function isAuth() { |
|  | return sessionStorage.getItem('authtoken') !== null; |
|  | } |
|  |  |
|  | function saveSession(res) { |
|  | sessionStorage.setItem('username', res.username); |
|  | sessionStorage.setItem('authtoken', res.\_kmd.authtoken); |
|  | sessionStorage.setItem('id', res.\_id); |
|  | } |
|  |  |
|  | function register(username, password) { |
|  | return kinvey.post('user', '', 'basic', { |
|  | username, |
|  | password |
|  | }) |
|  | } |
|  |  |
|  | function login(username, password) { |
|  | return kinvey.post('user', 'login', 'basic', { |
|  | username, |
|  | password |
|  | }); |
|  | } |
|  |  |
|  | function logout() { |
|  | return kinvey.post('user', '\_logout', 'kinvey'); |
|  | } |
|  |  |
|  | return { |
|  | register, |
|  | login, |
|  | logout, |
|  | saveSession, |
|  | isAuth |
|  | } |
|  | })(); |

//styles.notifications.css

|  |
| --- |
| #notifications { |
|  | float: right; |
|  | position: absolute; |
|  | top: 50px; |
|  | right: 0; |
|  | text-align: center; |
|  | margin: 1em; |
|  | } |
|  |  |
|  | .notification { |
|  | font-size: 1.5em; |
|  | color: #ffffff; |
|  | display: none; |
|  | text-align: right; |
|  | } |
|  |  |
|  | .notification > span { |
|  | display: inline-block; |
|  | padding: 0.5em 2em 0.5em 2em; |
|  | border-radius: 10px; |
|  | } |
|  |  |
|  | #loadingBox > span { |
|  | background: #7CB3E9; |
|  | } |
|  |  |
|  | #infoBox > span { |
|  | background: #393; |
|  | } |
|  |  |
|  | #errorBox > span { |
|  | background: #F50; |
|  | } |

//styles.site.css

|  |
| --- |
| body { |
|  | background: #333333; |
|  | } |
|  |  |
|  | .background-spotify { |
|  | width: 100%; |
|  | } |
|  |  |
|  | .background-image { |
|  | height: 100vh; |
|  | } |
|  |  |
|  | .new-song-btn { |
|  | background: white; |
|  | color: #333333; |
|  | text-transform: uppercase; |
|  | font-weight: 500; |
|  | margin-bottom: 34px; |
|  | border: 2px solid white; |
|  |  |
|  | } |
|  |  |
|  | .new-song-btn:hover { |
|  | cursor: pointer; |
|  | color: white; |
|  | background: #333333; |
|  | border: 2px solid white; |
|  | text-decoration: none !important; |
|  |  |
|  | } |
|  |  |
|  | .new-song-btn:hover a { |
|  | text-decoration: none !important; |
|  | } |
|  |  |
|  | h1 { |
|  | padding: 15px; |
|  | color: white; |
|  | } |
|  |  |
|  | .song { |
|  | width: 23%; |
|  | display: inline-block; |
|  | margin: 20px 5px; |
|  | background-color: white; |
|  | color: black; |
|  | text-align: center; |
|  | border: 3px solid white; |
|  | padding: 15px; |
|  | height: 387px; |
|  | vertical-align: top; |
|  | } |
|  |  |
|  | .song-container { |
|  | width: 74%; |
|  | margin: 20px auto auto; |
|  | padding: 50px 0; |
|  | min-height: 800px; |
|  |  |
|  | } |
|  |  |
|  | .btn-block { |
|  | display: block; |
|  | width: 32%; |
|  | margin-left: 34%; |
|  | font-weight: bold; |
|  | } |
|  |  |
|  | .cover { |
|  | height: 154px; |
|  | width: 90%; |
|  | border: 2px solid #333333; |
|  | padding: 0; |
|  | margin: 10px; |
|  | } |
|  |  |
|  | p { |
|  | font-style: italic; |
|  | margin-top: 0; |
|  | margin-bottom: 0; |
|  | font-weight: bold; |
|  | color: #333333; |
|  | } |
|  |  |
|  | h1, h4 { |
|  | text-align: center; |
|  | } |
|  |  |
|  | footer { |
|  | background-color: #f8f9fa !important; |
|  | } |
|  |  |
|  | form { |
|  | display: inline-block; |
|  | text-align: center; |
|  | margin-left: 40%; |
|  | } |
|  |  |
|  | a { |
|  | color: white; |
|  | text-decoration: none |
|  | } |
|  |  |
|  | .white-labels { |
|  | color: white; |
|  | text-align: left !important; |
|  | font-size: 1.3rem; |
|  | } |
|  |  |
|  | .btn-success { |
|  | background: #4B8B13; |
|  | } |
|  |  |
|  | .btn-danger { |
|  | background: #FF0100; |
|  | } |
|  |  |
|  | .add-link { |
|  | font-size: 1.5rem; |
|  | font-style: italic; |
|  | font-weight: 400; |
|  | text-decoration: underline; |
|  | } |
|  |  |
|  | form { |
|  | text-align: left; |
|  | font-size: 2rem; |
|  | margin-left: 33%; |
|  | width: 36%; |
|  | } |

//templates.home.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="homeView"> |
|  | <img class="m-auto background-image" width="100%" src="http://gomera24.com/wp-content/uploads/2017/06/566771.jpg"> |
|  | </section> |
|  | {{> footer}} |

//templates.login.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="loginView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>Login</h1> |
|  | <form action="#/login" method="POST"> |
|  | <div class="form-group"> |
|  | <label for="username" class="white-labels">Username</label> |
|  | <input id="username" type="text" name="username" class="form-control" placeholder="Enter username"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="password" class="white-labels">Password</label> |
|  | <input id="password" type="password" name="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary">Login</button> |
|  | </form> |
|  |  |
|  | <h4 class="mt-3 text-white">No account yet? <a href="#/register" class="add-link">Register</a></h4> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//templates.register.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="registerView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>Register</h1> |
|  | <form action="#/register" method="POST"> |
|  | <div class="form-group"> |
|  | <label for="username" class="white-labels">Username</label> |
|  | <input type="text" name="username" class="form-control" placeholder="Enter username"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="password" class="white-labels">Password</label> |
|  | <input type="password" name="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary">Register</button> |
|  | </form> |
|  | <h4 class="mt-3 text-white">Already have an account? <a href="#/login" class="add-link">Login</a></h4> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//templates.common.footer.hbs

|  |
| --- |
| <footer class="card-footer text-muted d-flex justify-content-center"> |
|  | &copy; SoftUni JS Core Apps Exams |
|  | </footer> |

//templates.common.header.hbs

|  |
| --- |
| <nav class="navbar navbar-expand-lg navbar-light bg-light fixed-top"> |
|  | <div class="collapse navbar-collapse" id="navbarText"> |
|  |  |
|  | {{#if isAuth}} |
|  | <ul class="navbar-nav mr-auto"> |
|  | <li class="nav-item active"> |
|  | <a class="nav-link" href="#/home">Home <span class="sr-only">(current)</span></a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/allSongs">All Songs</a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link " href="#/mySongs">My Songs</a> |
|  | </li> |
|  | </ul> |
|  |  |
|  | <ul class="navbar-nav justify-content-end"> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#">Welcome, {{username}}!</a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/logout">Logout</a> |
|  | </li> |
|  | </ul> |
|  |  |
|  | {{else}} |
|  | <ul class="navbar-nav mr-auto"></ul> |
|  |  |
|  | <ul class="navbar-nav justify-content-end"> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/login">Login</a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/register">Register</a> |
|  | </li> |
|  | </ul> |
|  | </div> |
|  | {{/if}} |
|  | </nav> |

//templates.song.AllSongs.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="allSongsView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>All Songs</h1> |
|  | <a href="#/createSong"> |
|  | <button type="button" class="btn-lg btn-block new-song-btn">Add a new song</button> |
|  | </a> |
|  |  |
|  | {{#each songs}} |
|  | {{> song}} |
|  | {{/each}} |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//templates.song.createSong.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="createSongView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>Create new song</h1> |
|  | <form action="#/createSong" method="POST"> |
|  | <div class="form-group"> |
|  | <label for="title" class="white-labels">Title</label> |
|  | <input id="title" type="text" name="title" class="form-control" placeholder="Title"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="artist" class="white-labels">Artist</label> |
|  | <input id="artist" type="text" name="artist" class="form-control" placeholder="Artist"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="imageURL" class="white-labels">imageURL</label> |
|  | <input id="imageURL" type="text" name="imageURL" class="form-control" placeholder="imageURL"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary">Create</button> |
|  | </form> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//templates.song.mySongs.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="mySongsView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>My Songs</h1> |
|  | {{#each songs}} |
|  | {{> song}} |
|  | {{/each}} |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//templates.song.song.hbs

|  |
| --- |
| <div class="song" id="{{id}}"> |
|  | <h5>Title: {{title}}</h5> |
|  | <h5>Artist: {{artist}}</h5> |
|  | <img class="cover" src="{{imageURL}}"/> |
|  |  |
|  | {{#if isCreator}} |
|  | <p>Likes: {{likeCounter}}; Listened {{listenCounter}} times</p> |
|  |  |
|  | <a href="#/remove/{{\_id}}"> |
|  | <button type="button" class="btn btn-danger mt-4">Remove</button> |
|  | </a> |
|  | <a href="#/listen/{{\_id}}"> |
|  | <button type="button" class="btn btn-success mt-4">Listen</button> |
|  | </a> |
|  | {{else}} |
|  | <p>Likes: {{likeCounter}}</p> |
|  |  |
|  | <a href="#/like/{{\_id}}"> |
|  | <button type="button" class="btn btn-primary mt-4">Like</button> |
|  | </a> |
|  | {{/if}} |
|  | </div> |

//not full solution from Hristomir Asenov

//index.html

|  |
| --- |
| <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <title>Spotify</title> |
|  | <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous"> |
|  | <link rel="stylesheet" href="./styles/notifications.css"> |
|  | <link rel="stylesheet" href="./styles/site.css"> |
|  |  |
|  | <script src="./scripts/notify.js"></script> |
|  | </head> |
|  |  |
|  | <body> |
|  | <div id="root"></div> |
|  |  |
|  | <div id="notifications"> |
|  | <div id="loadingBox" class="notification"> |
|  | <span>Loading …</span> |
|  | </div> |
|  | <div id="infoBox" class="notification"> |
|  | <span>Info</span> |
|  | </div> |
|  | <div id="errorBox" class="notification"> |
|  | <span>Error</span> |
|  | </div> |
|  | </div> |
|  |  |
|  | <script src="./node\_modules/jquery/dist/jquery.min.js"></script> |
|  | <script src="./node\_modules/handlebars/dist/handlebars.min.js"></script> |
|  | <script src="./node\_modules/sammy/lib/sammy.js"></script> |
|  | <script src="./node\_modules/sammy/lib/plugins/sammy.handlebars.js"></script> |
|  |  |
|  | <script src="./scripts/kinvey.js"></script> |
|  | <script src="./scripts/app.js"></script> |
|  | <script src="./scripts/notification.js"></script> |
|  | <!-- services --> |
|  | <script src="./scripts/services/user-service.js"></script> |
|  | <script src="./scripts/services/song-service.js"></script> |
|  |  |
|  | <!-- handlers --> |
|  | <script src="./scripts/handlers/user-handler.js"></script> |
|  | <script src="./scripts/handlers/home-handler.js"></script> |
|  | <script src="./scripts/handlers/song-handler.js"></script> |
|  | </body> |
|  |  |
|  | </html> |

//package.json

|  |
| --- |
| { |
|  | "dependencies": { |
|  | "handlebars": "^4.0.11", |
|  | "jquery": "^3.3.1", |
|  | "sammy": "^0.7.6" |
|  | } |
|  | } |

//scripts.app.js

|  |
| --- |
| const handlers = {} |
|  |  |
|  | $(() => { |
|  | const app = Sammy('#root', function () { |
|  | this.use('Handlebars', 'hbs'); |
|  | // home page routes |
|  |  |
|  | this.get('/index.html', handlers.getHome); |
|  | this.get('/', handlers.getHome); |
|  |  |
|  | this.get('#/home', handlers.getHome); |
|  |  |
|  | // user routes |
|  | this.get('#/register', handlers.getRegister); |
|  | this.get('#/login', handlers.getLogin); |
|  |  |
|  | this.post('#/register', handlers.registerUser); |
|  | this.post('#/login', handlers.loginUser); |
|  | this.get('#/logout', handlers.logoutUser); |
|  |  |
|  | // songs routes |
|  | this.get('#/allSongs', handlers.getAllSongs); |
|  | this.get('#/createSong', handlers.getCreateSong); |
|  | this.get('#/mySongs', handlers.getMySongs); |
|  | this.get('#/remove/:id', handlers.removeSong); |
|  | this.get('#/like/:id', handlers.likeSong); |
|  |  |
|  | this.post('#/createSong', handlers.createSong); |
|  |  |
|  | }); |
|  | app.run('#/home'); |
|  | }); |

//scripts.kinvey.js

|  |
| --- |
| const kinvey = (() => { |
|  | const BASE\_URL = 'https://baas.kinvey.com/'; |
|  | const APP\_KEY = ''; // <------------------- PUT YOUR APP KEY HERE |
|  | const APP\_SECRET = ''; // <------------------ PUT YOUR APP SECRET HERE |
|  |  |
|  | function makeAuth(auth) { |
|  | if (auth == 'basic') { |
|  | return { |
|  | 'Authorization': `Basic ${btoa(APP\_KEY + ':' + APP\_SECRET)}` |
|  | } |
|  | } else { |
|  | return { |
|  | 'Authorization': `Kinvey ${sessionStorage.getItem('authtoken')}` |
|  | } |
|  | } |
|  | } |
|  |  |
|  | function makeRequest(method, collection, endpoint, auth) { |
|  | return { |
|  | url: BASE\_URL + collection + '/' + APP\_KEY + '/' + endpoint, |
|  | method, |
|  | headers: makeAuth(auth) |
|  | } |
|  | } |
|  |  |
|  | function post(collection, endpoint, auth, data) { |
|  | let req = makeRequest('POST', collection, endpoint, auth); |
|  | req.data = data; |
|  | return $.ajax(req); |
|  | } |
|  |  |
|  | function get(collection, endpoint, auth, query) { |
|  |  |
|  | if(query){ |
|  | endpoint += query; |
|  | } |
|  |  |
|  | return $.ajax(makeRequest('GET', collection, endpoint, auth)); |
|  | } |
|  |  |
|  | function update(collection, endpoint, auth, data) { |
|  | let req = makeRequest('PUT', collection, endpoint, auth); |
|  | req.data = data; |
|  | return $.ajax(req); |
|  | } |
|  |  |
|  | function remove(collection, endpoint, auth) { |
|  | return $.ajax(makeRequest('DELETE', collection, endpoint, auth)); |
|  | } |
|  |  |
|  | return { |
|  | get, |
|  | post, |
|  | update, |
|  | remove |
|  | } |
|  | })() |

//scripts.notification.js

|  |
| --- |
| const notifications = (() => { |
|  | $(document).on({ |
|  | ajaxStart: () => $('#loadingBox').fadeIn(), |
|  | ajaxStop: () => $('#loadingBox').fadeOut() |
|  | }) |
|  |  |
|  | function showSuccess(message) { |
|  | let successBox = $('#infoBox'); |
|  | successBox.text(message); |
|  | successBox.fadeIn(); |
|  | successBox.fadeOut(3000); |
|  | } |
|  |  |
|  | function showError(message) { |
|  | let errorBox = $('#errorBox'); |
|  | errorBox.text(message); |
|  | errorBox.fadeIn(); |
|  | errorBox.fadeOut(3000); |
|  | } |
|  |  |
|  | return { |
|  | showSuccess, |
|  | showError |
|  | } |
|  | })() |

//scripts,notify.js

|  |
| --- |
| let notify = (() => { |
|  | $(document).on({ |
|  | ajaxStart: () => $('#loadingBox').show(), |
|  | ajaxStop: () => $('#loadingBox').fadeOut() |
|  | }) |
|  | function showInfo (message) { |
|  | let infoBox = $('#infoBox') |
|  | infoBox.find('span').text(message) |
|  | infoBox.fadeIn() |
|  | setTimeout(() => infoBox.fadeOut(), 3000) |
|  | } |
|  |  |
|  | function showError(message) { |
|  | let errorBox = $('#errorBox') |
|  | errorBox.find('span').text(message) |
|  | errorBox.fadeIn() |
|  | setTimeout(() => errorBox.fadeOut(), 3000) |
|  | } |
|  |  |
|  | function handleError (reason) { |
|  | showError(reason.responseJSON.description) |
|  | } |
|  |  |
|  | return { |
|  | showInfo, |
|  | showError, |
|  | handleError |
|  | } |
|  | })() |

//scripts.handlers.home-handler.js

|  |
| --- |
| handlers.getHome = function (ctx) { |
|  | ctx.isAuth = userService.isAuth(); |
|  | ctx.username = sessionStorage.getItem('username'); |
|  |  |
|  | ctx.loadPartials({ |
|  | header: '../views/common/header.hbs', |
|  | footer: '../views/common/footer.hbs' |
|  | }).then(function () { |
|  | this.partial('../views/home/homePage.hbs'); |
|  | }).catch(function (err) { |
|  | console.log(err); |
|  | }); |
|  | } |

//scripts.handlers.song-handler.js

|  |
| --- |
| handlers.getAllSongs = async function (ctx) { |
|  | ctx.isAuth = userService.isAuth(); |
|  | ctx.username = sessionStorage.getItem('username'); |
|  |  |
|  | try { |
|  | let songs = await songService.getAllSongs(); |
|  | let userId = sessionStorage.getItem('id'); |
|  | songs.forEach((song) => song.isCreator = song.\_acl.creator === userId); |
|  |  |
|  | ctx.songs = songs; |
|  |  |
|  | ctx.loadPartials({ |
|  | header: '../views/common/header.hbs', |
|  | footer: '../views/common/footer.hbs', |
|  | song: '../views/song/song.hbs' |
|  | }).then(function () { |
|  | this.partial('../views/song/allSongsPage.hbs'); |
|  | }).catch(function (err) { |
|  | console.log(err); |
|  | }); |
|  | } catch (e) { |
|  | console.log(e); |
|  | } |
|  | } |
|  |  |
|  | handlers.getCreateSong = function (ctx) { |
|  | ctx.isAuth = userService.isAuth(); |
|  | ctx.username = sessionStorage.getItem('username'); |
|  |  |
|  | ctx.loadPartials({ |
|  | header: '../views/common/header.hbs', |
|  | footer: '../views/common/footer.hbs' |
|  | }).then(function () { |
|  | this.partial('../views/song/createSongPage.hbs'); |
|  | }) |
|  | } |
|  |  |
|  | handlers.getMySongs = function (ctx) { |
|  |  |
|  | ctx.isAuth = userService.isAuth(); |
|  | ctx.username = sessionStorage.getItem('username'); |
|  |  |
|  | songService.getAllMySongs().then(function (res) { |
|  |  |
|  | let userId = sessionStorage.getItem('id'); |
|  | res.forEach((song) => song.isCreator = song.\_acl.creator === userId); |
|  |  |
|  | ctx.songs = res; |
|  |  |
|  | ctx.loadPartials({ |
|  | header: '../views/common/header.hbs', |
|  | footer: '../views/common/footer.hbs', |
|  | song: '../views/song/song.hbs' |
|  | }).then(function () { |
|  | this.partial('../views/song/mySongsPage.hbs'); |
|  | }) |
|  | }) |
|  | } |
|  |  |
|  | handlers.createSong = function (ctx) { |
|  |  |
|  | let data = { ...ctx.params, likeCounter: 0, listenCounter: 0 }; |
|  |  |
|  | if(data.title.length < 6) { |
|  | notifications.showError('The title should be at least 6 characters long!'); |
|  | } else if (data.artist.length < 3) { |
|  | notifications.showError('The artist should be at least 3 characters long!'); |
|  | } else if (!data.imageURL.startsWith('http')){ |
|  | notifications.showError('The image should start with "http://" or "https://"'); |
|  | } else { |
|  |  |
|  | songService.createSong(data).then(function (res) { |
|  | notifications.showSuccess('Song created successfully!'); |
|  | ctx.redirect('#/mySongs'); |
|  | }).catch(function (err) { |
|  | console.log(err); |
|  | }) |
|  | } |
|  | } |
|  |  |
|  | handlers.removeSong = function (ctx) { |
|  |  |
|  | songService.removeSong(ctx.params.id).then(function () { |
|  | notifications.showSuccess('Song was removed successfully!'); |
|  | ctx.redirect('#/mySongs'); |
|  | }).catch(function (err) { |
|  | console.log(err); |
|  | }) |
|  | } |
|  |  |
|  | handlers.likeSong = async function (ctx) { |
|  |  |
|  | let id = ctx.params.id; |
|  |  |
|  | try { |
|  | let song = await songService.getASong(id); |
|  | let newLikes = Number(song.likeCounter) + 1; |
|  | song.likeCounter = newLikes; |
|  |  |
|  | songService.likeSong(id, song).then(function () { |
|  | notifications.showSuccess('Song was liked successfully!'); |
|  | ctx.redirect('#/allSongs'); |
|  | }).catch(function (err) { |
|  | console.log(err); |
|  | }) |
|  | } catch (e) { |
|  | console.log(e); |
|  | } |
|  | } |

//scripts.handlers.user-handler.js

|  |
| --- |
| handlers.getRegister = function (ctx) { |
|  |  |
|  | ctx.loadPartials({ |
|  | header: '../views/common/header.hbs', |
|  | footer: '../views/common/footer.hbs' |
|  | }).then(function () { |
|  | this.partial('../../views/user/registerPage.hbs'); |
|  | }).catch(function (err) { |
|  | console.log(err); |
|  | }); |
|  | } |
|  |  |
|  | handlers.getLogin = function (ctx) { |
|  | ctx.loadPartials({ |
|  | header: '../views/common/header.hbs', |
|  | footer: '../views/common/footer.hbs' |
|  | }).then(function () { |
|  | this.partial('../../views/user/loginPage.hbs'); |
|  | }).catch(function (err) { |
|  | console.log(err); |
|  | }); |
|  | } |
|  |  |
|  | handlers.registerUser = function (ctx) { |
|  | let username = ctx.params.username; |
|  | let password = ctx.params.password; |
|  |  |
|  | userService.register(username, password).then((res) => { |
|  | userService.saveSession(res); |
|  | notifications.showSuccess('User registered successfully'); |
|  | ctx.redirect('#/home'); |
|  | }).catch(function (err) { |
|  | notifications.showError(err.responseJSON.description); |
|  | }); |
|  | } |
|  |  |
|  | handlers.logoutUser = function (ctx) { |
|  | userService.logout().then(() => { |
|  | sessionStorage.clear(); |
|  | notifications.showSuccess('User logged out successfully'); |
|  | ctx.redirect('#/home'); |
|  | }) |
|  | } |
|  |  |
|  | handlers.loginUser = function (ctx) { |
|  | let username = ctx.params.username; |
|  | let password = ctx.params.password; |
|  | userService.login(username, password).then((res) => { |
|  | userService.saveSession(res); |
|  | notifications.showSuccess('User logged in successfully'); |
|  | ctx.redirect('#/home'); |
|  | }).catch(function (err) { |
|  | notifications.showError(err.responseJSON.description); |
|  | }); |
|  | } |

//scripts.services.song-service.js

|  |
| --- |
| const songService = (() => { |
|  |  |
|  | function createSong (data) { |
|  | return kinvey.post('appdata', 'songs', 'kinvey', data); |
|  | } |
|  |  |
|  | function getAllSongs () { |
|  | return kinvey.get('appdata', 'songs', 'kinvey', '?query={}&sort={"likeCounter": -1}'); |
|  | } |
|  |  |
|  | function getAllMySongs () { |
|  | return kinvey.get('appdata', 'songs', 'kinvey', `?query={"\_acl.creator":"${sessionStorage.getItem('id')}"}`); |
|  | } |
|  |  |
|  | function removeSong (id) { |
|  | return kinvey.remove('appdata', `songs/${id}`, 'kinvey'); |
|  | } |
|  |  |
|  | function likeSong (id, song) { |
|  | return kinvey.update('appdata', `songs/${id}`, 'kinvey', song); |
|  | } |
|  |  |
|  | function getASong (id) { |
|  | return kinvey.get('appdata', `songs/${id}`, 'kinvey'); |
|  | } |
|  |  |
|  | return { |
|  | createSong, |
|  | getAllSongs, |
|  | getAllMySongs, |
|  | removeSong, |
|  | likeSong, |
|  | getASong |
|  | } |
|  | })(); |

//scripts.services.user-service.js

|  |
| --- |
| const userService = (() => { |
|  | function isAuth() { |
|  | return sessionStorage.getItem('authtoken') !== null; |
|  | } |
|  |  |
|  | function saveSession(res) { |
|  | sessionStorage.setItem('username', res.username); |
|  | sessionStorage.setItem('authtoken', res.\_kmd.authtoken); |
|  | sessionStorage.setItem('id', res.\_id); |
|  | } |
|  |  |
|  | function register(username, password) { |
|  | return kinvey.post('user', '', 'basic', { |
|  | username, |
|  | password |
|  | }) |
|  | } |
|  |  |
|  | function login(username, password) { |
|  | return kinvey.post('user', 'login', 'basic', { |
|  | username, |
|  | password |
|  | }); |
|  | } |
|  |  |
|  | function logout() { |
|  | return kinvey.post('user', '\_logout', 'kinvey'); |
|  | } |
|  |  |
|  | return { |
|  | register, |
|  | login, |
|  | logout, |
|  | saveSession, |
|  | isAuth |
|  | } |
|  | })() |

//styles.notifications.css

|  |
| --- |
| #notifications { |
|  | float: right; |
|  | position: absolute; |
|  | top: 50px; |
|  | right: 0; |
|  | text-align: center; |
|  | margin: 1em; |
|  | } |
|  |  |
|  | .notification { |
|  | font-size: 1.5em; |
|  | color: #ffffff; |
|  | display: none; |
|  | text-align: right; |
|  | } |
|  |  |
|  | .notification>span { |
|  | display: inline-block; |
|  | padding: 0.5em 2em 0.5em 2em; |
|  | border-radius: 10px; |
|  | } |
|  |  |
|  | #loadingBox{ |
|  | background: #7CB3E9; |
|  | } |
|  | #infoBox{ |
|  | background: #393; |
|  | } |
|  | #errorBox{ |
|  | background: #F50; |
|  | } |

//styles.site.css

|  |
| --- |
| body { |
|  | background: #333333; |
|  | } |
|  |  |
|  | .background-spotify { |
|  | width: 100%; |
|  | } |
|  |  |
|  | .background-image { |
|  | height: 100vh; |
|  | } |
|  |  |
|  | .new-song-btn { |
|  | background: white; |
|  | color: #333333; |
|  | text-transform: uppercase; |
|  | font-weight: 500; |
|  | margin-bottom: 34px; |
|  | border: 2px solid white; |
|  |  |
|  | } |
|  |  |
|  | .new-song-btn:hover { |
|  | cursor: pointer; |
|  | color: white; |
|  | background: #333333; |
|  | border: 2px solid white; |
|  | text-decoration: none !important; |
|  |  |
|  | } |
|  |  |
|  | .new-song-btn:hover a { |
|  | text-decoration: none !important; |
|  | } |
|  |  |
|  | h1 { |
|  | padding: 15px; |
|  | color: white; |
|  | } |
|  |  |
|  | .song { |
|  | width: 23%; |
|  | display: inline-block; |
|  | margin: 20px 5px; |
|  | background-color: white; |
|  | color: black; |
|  | text-align: center; |
|  | border: 3px solid white; |
|  | padding: 15px; |
|  | height: 387px; |
|  | vertical-align: top; |
|  | } |
|  |  |
|  | .song-container { |
|  | width: 74%; |
|  | margin: 20px auto auto; |
|  | padding: 50px 0; |
|  | min-height: 800px; |
|  |  |
|  | } |
|  |  |
|  | .btn-block { |
|  | display: block; |
|  | width: 32%; |
|  | margin-left: 34%; |
|  | font-weight: bold; |
|  | } |
|  |  |
|  | .cover { |
|  | height: 154px; |
|  | width: 90%; |
|  | border: 2px solid #333333; |
|  | padding: 0; |
|  | margin: 10px; |
|  | } |
|  |  |
|  | p { |
|  | font-style: italic; |
|  | margin-top: 0; |
|  | margin-bottom: 0; |
|  | font-weight: bold; |
|  | color: #333333; |
|  | } |
|  |  |
|  | h1, h4 { |
|  | text-align: center; |
|  | } |
|  |  |
|  | footer { |
|  | background-color: #f8f9fa !important; |
|  | } |
|  |  |
|  | form { |
|  | display: inline-block; |
|  | text-align: center; |
|  | margin-left: 40%; |
|  | } |
|  |  |
|  | a { |
|  | color:white; |
|  | text-decoration:none |
|  | } |
|  |  |
|  | .white-labels { |
|  | color: white; |
|  | text-align: left !important; |
|  | font-size: 1.3rem; |
|  | } |
|  |  |
|  | .btn-success { |
|  | background: #4B8B13; |
|  | } |
|  |  |
|  | .btn-danger { |
|  | background: #FF0100; |
|  | } |
|  |  |
|  | .add-link { |
|  | font-size: 1.5rem; |
|  | font-style: italic; |
|  | font-weight: 400; |
|  | text-decoration: underline; |
|  | } |
|  |  |
|  | form { |
|  | text-align: left; |
|  | font-size: 2rem; |
|  | margin-left: 33%; |
|  | width: 36%; |
|  | } |

//views.common.footer.hbs

|  |
| --- |
| <footer class="card-footer text-muted d-flex justify-content-center"> |
|  | &copy; SoftUni JS Core Apps Exams |
|  | </footer> |

//views.common.header.hbs

|  |
| --- |
| <nav class="navbar navbar-expand-lg navbar-light bg-light fixed-top"> |
|  | <div class="collapse navbar-collapse" id="navbarText"> |
|  | <ul class="navbar-nav mr-auto"> |
|  | {{#if isAuth}} |
|  | <li class="nav-item active"> |
|  | <a class="nav-link" href="#/home">Home <span class="sr-only">(current)</span></a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/allSongs">All Songs</a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link " href="#/mySongs">My Songs</a> |
|  | </li> |
|  | <ul class="navbar-nav justify-content-end"> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#">Welcome, {{username}}</a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/logout">Logout</a> |
|  | </li> |
|  | </ul> |
|  | {{else}} |
|  | <ul class="navbar-nav mr-auto"></ul> |
|  | <ul class="navbar-nav justify-content-end"> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/login">Login</a> |
|  | </li> |
|  | <li class="nav-item"> |
|  | <a class="nav-link" href="#/register">Register</a> |
|  | </li> |
|  | </ul> |
|  | {{/if}} |
|  | </ul> |
|  | </div> |
|  | </nav> |

//views.home.homePage.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="homeView"> |
|  | <img class="m-auto background-image" width="100%" src="http://gomera24.com/wp-content/uploads/2017/06/566771.jpg"> |
|  | </section> |
|  | {{> footer}} |

//views.song.allSongsPage.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="allSongsView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>All Songs</h1> |
|  | <a href="#/createSong"> |
|  | <button type="button" class="btn-lg btn-block new-song-btn">Add a new song</button> |
|  | </a> |
|  | {{#each songs}} |
|  | {{> song}} |
|  | {{/each}} |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//views.song.createSongPage.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="createSongView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>Create new song</h1> |
|  | <form action="#/createSong" method="POST"> |
|  | <div class="form-group"> |
|  | <label for="title" class="white-labels">Title</label> |
|  | <input id="title" type="text" name="title" class="form-control" placeholder="Title"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="artist" class="white-labels">Artist</label> |
|  | <input id="artist" type="text" name="artist" class="form-control" placeholder="Artist"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="imageURL" class="white-labels">imageURL</label> |
|  | <input id="imageURL" type="text" name="imageURL" class="form-control" placeholder="imageURL"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary">Create</button> |
|  | </form> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//views.song.mySongsPage.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="mySongsView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>My Songs</h1> |
|  | {{#each songs}} |
|  | {{> song}} |
|  | {{/each}} |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//views.song.song.hbs

|  |
| --- |
| <div class="song" > |
|  | <h5>Title: {{title}}</h5> |
|  | <h5>Artist: {{artist}}</h5> |
|  | <img class="cover" src="{{imageURL}}" /> |
|  | {{#if isCreator}} |
|  | <p>Likes: {{likeCounter}}; Listened {{listenCounter}} times</p> |
|  | <a href="#/remove/{{\_id}}"><button type="button" class="btn btn-danger mt-4">Remove</button></a> |
|  | <a href="#"><button type="button" class="btn btn-success mt-4">Listen</button></a> |
|  | {{else}} |
|  | <p>Likes: {{likeCounter}}</p> |
|  | <a href="#/like/{{\_id}}"><button type="button" class="btn btn-primary mt-4">Like</button></a> |
|  | {{/if}} |
|  | </div> |

//views.user.loginPage.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="loginView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>Login</h1> |
|  | <form action="#/login" method="POST"> |
|  | <div class="form-group"> |
|  | <label for="username" class="white-labels">Username</label> |
|  | <input id="username" type="text" name="username" class="form-control" placeholder="Enter username"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="password" class="white-labels">Password</label> |
|  | <input id="password" type="password" name="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary">Login</button> |
|  | </form> |
|  |  |
|  | <h4 class="mt-3 text-white">No account yet? <a href="#/register" class="add-link">Register</a></h4> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |

//views.user.registerPage.hbs

|  |
| --- |
| {{> header}} |
|  | <section id="registerView"> |
|  | <div class="background-spotify"> |
|  | <div class="song-container"> |
|  | <h1>Register</h1> |
|  | <form action="#/register" method="POST"> |
|  | <div class="form-group"> |
|  | <label for="username" class="white-labels">Username</label> |
|  | <input type="text" name="username" class="form-control" placeholder="Enter username"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="password" class="white-labels">Password</label> |
|  | <input type="password" name="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary">Register</button> |
|  | </form> |
|  | <h4 class="mt-3 text-white">Already have an account? <a href="#/login" class="add-link">Login</a></h4> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | {{> footer}} |