

Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика



Тема: Sportsemblr - уеб приложение за
организиране на спортни събития

Дисциплина: **Fullstack Application Development with Node.js +
Express.js + React.js**

Автори:

Иван Младенов, ф.н. 61950, СИ, 3 курс

Бранимир Кирилов, ф.н. 61908, СИ, 3 курс

1. Описание

Sportsemlr.2 помага на хората, които обичат да спортуват да намерят съотборници или опоненти, с които да играят заедно. Най-голямата причина за това, че хората не спортуват достатъчно е липсата на достатъчно голямо общество със същите интереси. Със Sportsemlr.2 всеки, който желае да спортува, може просто да добави ново спортно събитие с локацията, датата и часа на срещата, за да могат другите потребители да видят и да се присъединят. След като някой се присъедини, той може да пише в чата на събитието за допълнителни детайли или доизясняване. След всяка игра, участниците могат да споделят мнението си за другите играчи, за по-добро и приятно събиране следващия път. Потребителите могат да добавят също така и игрища, които притежават или управляват, за да могат да организират спортни събития там. Администраторите на системата се грижат за доброто пребиваване на потребителите. Те имат пълен контрол върху събитията, терените и потребителите, за да се спазва добрия тон на игра. Различните видове спортове също се управляват от администраторите.

2. Видове потребители

Основните видове потребители са:

Анонимен потребител (гост)

- Разглежда списъците от спортни събития
- Разглежда детайлите на спортно събитие
- Разглежда списъците от терени
- Разглежда детайлите на терен
- Регистрира се

Регистриран потребител

- Вписва се в системата
- Отписва се
- Редактира информацията в профила си
- Създава събития
- Редактира събития, които е създал
- Одобрява или отхвърля заявки за участие в събитие
- Пише и модерират чата на събитие, което е създал
- Заявява участие в събитие
- Пише в чата на събитие, в което участва
- Създава терени
- Редактира и трие терени, които е добавил
- Остава мнение за събития, в които е участвал

Администратор

- Разглежда всички събития, терени, спортове и потребители на системата
- Редактира събития

- Изтрива събития
- Редактира терени
- Изтрива терени
- Редактира спортове
- Изтрива потребители

3. Изисквания към системата

○ Функционални изисквания

- Да поддържа три роли за потребители - гост, регистриран потребител и администратор
- Гостът може само да разглежда информацията за терените в системата, без да добавя нова или да редактира текущата
- Всеки потребител трябва да може лесно да намира информацията за събитие, което търси.
- Всеки потребител трябва да може лесно да търси събитията по вид спорт
- Гостът може само да разглежда информацията за събитията в системата, без да добавя нова или да редактира текущата
- Всеки потребител трябва да може лесно да намира информацията за терен, който търси.
- Всеки потребител трябва да може да разглежда информацията в профила на друг потребител
- Всеки потребител трябва да може да се регистрира в системата с уникален email
- Всеки регистриран потребител трябва да може да влиза в системата с email и парола
- Всеки регистриран потребител може да създава събитие
- Всеки регистриран потребител може да редактира информацията за събитие, което е създал
- Всеки регистриран потребител може да пише в чата на събитие, което е създал
- Всеки регистриран потребител може да одобрява или отхвърля заявки за участие в събитие, което е създал
- Всеки регистриран потребител може да изпраща заявка за включване в събитие
- Всеки регистриран потребител може да пише в чата на събитие, в което участва
- Всеки регистриран потребител може да напише ревю на събитие, в което е участвал и е свършило
- Всеки регистриран потребител може да редактира информацията за профила си
- Всеки регистриран потребител вижда всички събития, в които е участвал или участва
- Администраторите на системата имат достъп до всички функционалности

- Администраторът може да добавя, променя и трие списъка със спортове
- Администраторът може да дава администраторски права на даден потребител
- Администраторът може да премахва администраторски права на даден потребител
- Администраторът може да трие потребител

○ Нефункционални изисквания

- Потребителският интерфейс трябва да е интуитивен и лесен за ползване
- При записването на паролата в базата, тя не трябва да се пази като текст, а да бъде хеширана
- Аутентикацията трябва да се реализира с JSON Web Token
- Системата трябва да е реализирана с клиент-сървър архитектура

4. Използвани технологии и библиотечни модули

○ Клиент

Компонентите, използвани за разработката на клиентската част на приложението, са:

- React
- ReactRouter
- Redux
- Babel
- Webpack - module bundler
- Bootstrap
- Font-Awesome
- Jquery
- Toastr
- Eslint

○ Сървър

Компонентите, използвани за разработката на сървърната част на приложението, са:

- Node
- Express
- Passport
- MongoDB
- Mongoose

- CryptoJs
- Body-parser

5. Взаимодействие между основните компоненти

- ❖ Events
 - EventList използва EventShort, ManageEvent, VisibleFilters, MapComponent, TextInput
 - EventDetails използва EventReview, EventRequest
- ❖ Visible Filters
 - Visible Filters използва ActiveFilter, GenericFilter
- ❖ Place
 - PlacesList използва PlacesShort, ManagePlace, PlaceForm
- ❖ Profile
 - Profile използва UpdateProfileModal
- ❖ Admin
 - Admin използва ManageSports, ManageUsers
 - ManageSports използва SportsList
 - SportsList използва Sport
 - ManageUsers използва UsersList
- ❖ App
 - App използва Aside, ExtendedNavbar, Notification

6. Основни изгледи

Име на изгледа	Описание	URL
Home	Начална страница	/
Events List	Списък с събитията	/events
Event Details	Информация за едно събитие	/event/:id
Manage (Create) Event	Редактиране на събитие	/events (modal)
Places List	Списък с места	/places
Place Details	Информация за едно място	/place/:id
Profile	Информация за профила	/profile
Login	Екран за вписване	/login
Register	Екран за регистрация	/register

Users	Списък с потребителите	/users
User Details	Информация за един потребител	/users/:id
(Admin) Manage Sports	Управление спортове	/admin/sports
(Admin) Manage Users	Управление потребители	/admin/users
Admin Dashboard	Начало за админ	/admin

7. Приложно програмен интерфейс на сървъра

Име на ресурс	Описание	Метод	URI	Аутентикация
Login	Вписва потребител в системата	POST	/login	Няма
Event	Създаване	POST	/events	Да
	Списък с всички	GET	/events	Няма
	Активни събития	GET	/events/active	Не
	Редакция на събитие	PUT	/events/:id	Да
	Информация за едно събитие	GET	/events/:id	Да
	Изтриване на събитие	DELETE	/events/:id	Да
Place	Създаване	POST	/places	Да

	Списък с всички	GET	/places	Няма
	Редакция на място	PUT	/places/:id	Да
	Изтриване на място	DELETE	/places/:id	Да
Request	Всички заявки за добавяне към събитие	POST	events/:id/requests	Да
	Отговор на заявки	PUT	events/:id/:requestId	Да
Review	Добавяне на мнение	POST	events/:id/review	Да
Users	Създаване на потребител	POST	/users	Не
	Списък с потребителите	GET	/users	Не
	Редакция на потребител	PUT	/users/:id	Да
	Създаване на админ	PUT	/users/:id/admin	Да
	Информация за един потребител	GET	/users/:id	Да
	Изтриване на профил	DELETE	/users/:id	Да

8. Инсталиране и конфигуриране на системата

- Инсталирайте най-новите версии на node.js, mongodb и npm.
- Клонирайте git репозиторията - `git clone https://github.com/lordendbear/Sportsembler.2.git`
- Инсталирайте пакетите на сървърното приложение - `cd server | npm install`
- Инсталирайте пакетите на клиентското приложение - `cd client | npm install`

- Стартирайте mongodb сървъра - `mongod`
- Стартирайте приложенията - `npm start`

*За тестване на администраторските функционалности трябва ръчно да се промени ролята на регистрирания потребител в базата да бъде "Admin".

9. Потребителска документация за работа със системата

Основна част от навигацията в приложението се извършва през навигационната лента в горната част. Тя предлага достъп до основните страници, предназначени за потребителя. Всички потребители имат достъп до страницата със събития и терени. На лентата на анонимния потребител има линк за регистрация и за вписване в системата. Регистрираният потребител има достъп до профила си и линк за изход. Администратора има допълнителен линк за връзка с администраторския панел.

Всяка страница, съдържаща списък от елементи, съдържа също и линк към страницата с повече информация за даден елемент(събитие, терен, потребител). Функционалностите на тези страници са ограничени в зависимост от ролята на текущия потребител.

Регистрираният потребител има възможност за повече дейности, свързани със събития и терени. Също така може да променя и профила си.

Администраторът може да извършва всички дейности, които регистрираният потребител. Допълнително има достъп до администраторския панел, в който може да управлява събития, терени, потребители и спортове

10. Бъдещо развитие на системата

○

11. Използвани материали

[1.] Документация на mongoose - <http://mongoosejs.com/docs/>

[2.] Документация на Express - <https://expressjs.com/en/4x/api.html>

[3.] Документация на React - <https://reactjs.org/>