

owLearning

**Факултет по математика и информатика**

Разработка на клиент-сървър (fullstack) приложения с Node.js + Express.js + React.js

**Автори:**

Полина Кръстева 61789,

Мирослав Лалев 61844

1. **Основни цели и задачи**

Интернет епохата оказва голямо влияние на хората - такова че те са го интегрирали във всяка една част от живота си и го използват непрекъснато за най-различни цели. Една от най-важните, от които е обучението. Днес всеки един от нас може да открие всичко, за което се интересува за части от секундата, за разлика от хората от миналото, които не са разполагали с толкова обемна литература, каквато се оказва Интернет. В наши дни има огромно разнообразие от онлайн документи за електронно обучение. Тези документи се намират в прости уеб страници или в по-големи платформи, но това затруднява потребителя при ориентирането му в търсене на правилния и напълно достоверен източник на знания. Затова чрез нашата платформа ние ще се стремим да създадем достоверна източник на информация, която ще бъде представяна под формата на онлайн курсове, които потребителят може да записва и да намира отговори на въпросите си. Курсовете ще бъдат от най-различно естество - наука, изкуство, литература и други. Така всеки един от потребителите може да избере насоката, в която иска да се развива. Основната цел на нашия проект и това, което ще го различи от множеството платформи за обучение е че курсовете ще бъдат качвани от преподаватели. Също така бихме си кооперирали с университети и училища, което допълнително би увеличило увереността в автентичността на нашата система. Платформата ще поддържа различни типове потребители с различни нива на достъп и фунционалности, за които има повече подробности в точка 2.

1. **Целева група**

Проектът е насочен към две основни целеви групи – учители и ученици. Пред всеки потребител има възможността да използва системата като:

* Нерегистриран потребител(достъп до системата по подразбиране) - може да се регистрира и преглежда списък с курсовете
* Регистриран потребител - ученик - може да разглежда курсове и да се записва в тях, да преглежда материалите, комуникира с останалите чрез коментари, да следи статуса на курса и да оценява
* Регистриран потребител - преподавател - може да администрира курсове - да добавя, премахва курсове и материали в тях, да следи статуса на курсовете си

1. **Изисквания към системата**
   1. **функционални**

* Регистрация - потребителите ще имат възможност да се регистрират в системата, като ще им бъде даден избор като какви да се регистрират - учител или ученик.
* Управление на профил - потребителите ще могат да преглеждат и променят профила си по собствено желание.
* Добавяне на курс - потребителите регистрирани като учители ще могат да добавят нови собствени курсове
* Записване/Абониране за курс - всеки потребител може да се запише за даден курс и да ползва материалите напълно свободно
* Оценка на курсове - потребител, регистрирани като ученик ще могат да дава своята оценка за даден курс
* Обратна връзка за курс – учениците ще могат да оставят коментари и мнения към курса и преподавателя му
* Преглед на каталог на всички налични курсове – потребителят може да разглежда списък от всички курсове в платформата
* Преглед на определена категория курсове – потребителят може да филтрира списъка от всички курсове по определени параметри
* Преглед на урок от даден курс – ученикът, абониран за даден курс, може да преглежда съдържанието на конкретен урок от курса
* Коментиране на урок – учениците могат да оставят свободно своите коментари за даден урок – тази част може да бъде използвана като място за задаване на въпроси или коментиране на дадена тема от урока
* Известяване на клиент за настъпване на определени събития – например действия с курсове към които е абониран
* Добавяне на нов урок – учителите могат да добавят нови уроци към своите курсове, било то текст, видео или външни препратки
* Добавяне на условие на домашно – извършвано от учителя към курс, който той админстрира
* Добавяне на решение на домашно – извършвано от ученика за курс, за който той е абониран
* Оценка на домашно – извършва се от учителя за решение, предадено към курс, който той администрира
  1. **Нефункционални**
* Производителност - системата трябва да връща огговор в рамките на няколко секунди и също така трябва да може да издържа натоварване от 10 000 потребители.
* Използваемост - системата трябва да е колкото се може по-удобна за потребителите. Не трябва да отнеме на някого повече от 30 минути да се запознае със всички дейности, които може да извърши.
* Сигурност - системата трябва да предлага високо ниво на сигурност, тъй като потребителите ще използват свои лични данни, които не трябва да бъдат достъпвани от други лица.
* Разположение - системата ще е на разположение постоянно, но ще има определени часове за подръжка
* Преносимост - системата трябва да бъде достъпна от всеки от по-известните и широко употребявани браузъри и версии

1. **Използвани технологии**

* Front end частта е реализирана чрез React.js и React Bootstrap
* Back end частта използва Node.js, Express.js, за базата данни – pg-promise, node-pg-migrate, a за authentication – Passport.js

1. **REST API**

Налични са няколко основни REST endpointa:

* **/users** – всички заявки свързани с потребители

Създаването на нов потребител става с POST на този адрес с body от вида:

{

“email”: “email”,

“firtsname”: “firstname”,

“lastname”: “lastname”,

“password”: “password”,

“role”: “STUDENT”

}

Също така, може да се взима и обновява информация за конкретен потребител на **/users/:id**

* **/courses**

Тук могат да се вземат всички курсове, да се създаде нов курс, което задължително изисква потребителят да е логнат:

{

“name”: ”name”,

“description”: “description”

}

Информацията за отделен курс е на **/courses/:id**

Нa **/courses/:id/lectures** може да се качва нова лекция:

{

“name”: ”name”,

“description”: “description”

}

Информацията за отделна лекция е на **/courses/:courseId/lectures/:lectureId**