|  |  |  |
| --- | --- | --- |
| https://lh5.googleusercontent.com/-TUW1zMeNCdBTxjQj8wCfZLxcCNEkPmDZevTp_XZNmBdC6rXeDs-0WqArLGkA4BWO0gW4DfizRlCebuzSzX0AClIIPDEucPQFcMQEFyyRRptmePLbPyhT1dfOJOwPf3BwEwDTnLA | **Софийски университет**  **„Св. Климент Охридски“**  Факултет по математика и информатика |  |

**КУРСОВА РАБОТА**

# “Project management system”

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

**Изготвили**:

Камен Тодоров, 61808

Петър Татерски, 61845

## Функционални и нефункционални изисквания

* 1. Create project - Registered user will be able to create project to which he can add tasks and team members.
  2. Manage project info - Edit project information
  3. Browse project - List own projects and projects in which the user is member
  4. Create task - Create user task with detailed information about the task - time frame, description, task name, checklist, tags etc.
  5. Create subtask - Create subtask to a task
  6. Edit task info - Edit already created task information
  7. Browse tasks for project - See all tasks that are related to project
  8. Browse members for project - View all members of a project.
  9. Add/remove member - Add or remove existing user as a member to the project.
  10. Register - Anonymous User can register in the system by providing a valid e-mail address, first and last name, and choosing password. By default, all new registered users have Regular role.  
      Administrator can register new user by entering User Data and choosing a Role (Regular or Administrator).
  11. Change User Data - Registered User can view and edit their personal User Data.  
      Administrator can view and edit User Data of all Users and assign them Roles: Regular or Administrator.
  12. Manage Users - Administrator can browse users.  
      Administrator can choose a User to manage and can manage the chosen User - edit or delete.  
      Administrator can create a new user.
  13. Create sprint - Groups tasks in sprint
  14. Finish sprint - Complete sprint (The sprint must not have pending tasks. All open tasks will go in the backlog - list of all tasks that doesn’t have sprint)
  15. List all sprints - Show all sprints to the user
  16. Move task to sprint - Move task to sprint, the sprint should be active
  17. View sprint statistic - View details about the sprint - open/closed/opened tasks, time
  18. Estimation from the team
  19. Comments for the estimations

## Описание на използваните технологии

* 1. Front-end
     1. React.js – JavaScript библиотека за създаване на потребителски интерфейс.
     2. React-router-dom – библиотека за рутиране в SPA
     3. Axios – Prommised базиран HTTP клиент за браузъри и node.js
     4. Babel – JavaScript компилатор за бъдещ JavaScript
     5. Webpack – Module bundler
     6. Bootstrap – Библиотека с отворен код, съдържаща css стилове, за лесно създаване на приложения с HTML, CSS и JavaScript
  2. Back-end
     1. express - web framework for node.js
     2. indicative – библиотека за валидация
     3. passport – библиотека за ауторизация на потребители
     4. sqlite3 – библиотека за връзка с база данни sqlite

## Архитектура, реализация и взимодействие между компоненти

Архитектурата е от тип client-server. Клиентът е реализиран на react.js, а сървърът – на node.js.

Клиентът отправя заявки към сървъра с помощта на axios по HTTP/1.1 протокола. Сървърът консумира заяваките посредством REST API.

## Интерфейси

* 1. /api/projects - GET - get all projects for which the user is owner or is a team member. POST - create project with information about it
  2. /api/projects/:projectId - GET - get information about given project PUT - update project data, DELETE - delete project from the system
  3. /api/projects/:projectId/members - GET - get all project members
  4. /api/projects/:projectId/members/:userId - POST - add user as a member the project, DELETE - remove member from project
  5. /api/tasks/:taskId - GET - get detailed task information, PUT - update task information, POST - create subtask
  6. /api/tasks/project/:projectId - GET - get all project tasks from the backlog, POST - create task for given project with needed basic information in the backlog
  7. /api/tasks/sprint/:spirntId - GET - get tasks for the sprint
  8. /api/tasks/sprint/:spirntId/project/:projectId – POST - create task for the sprint
  9. /api/users - GET User Data for all users, and POST new User Data. Available only for Administrators
  10. /api/users/:userId - GET, PUT, DELETE User Data for User with specified userId
  11. /api/login - POST - authenticate user

## Инсталиране

Предусловия:

npm v3.10.10 or above

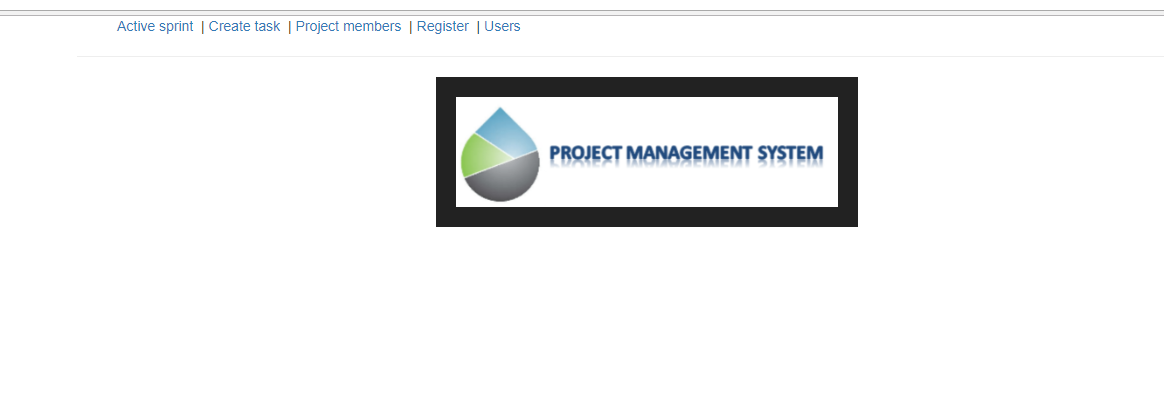
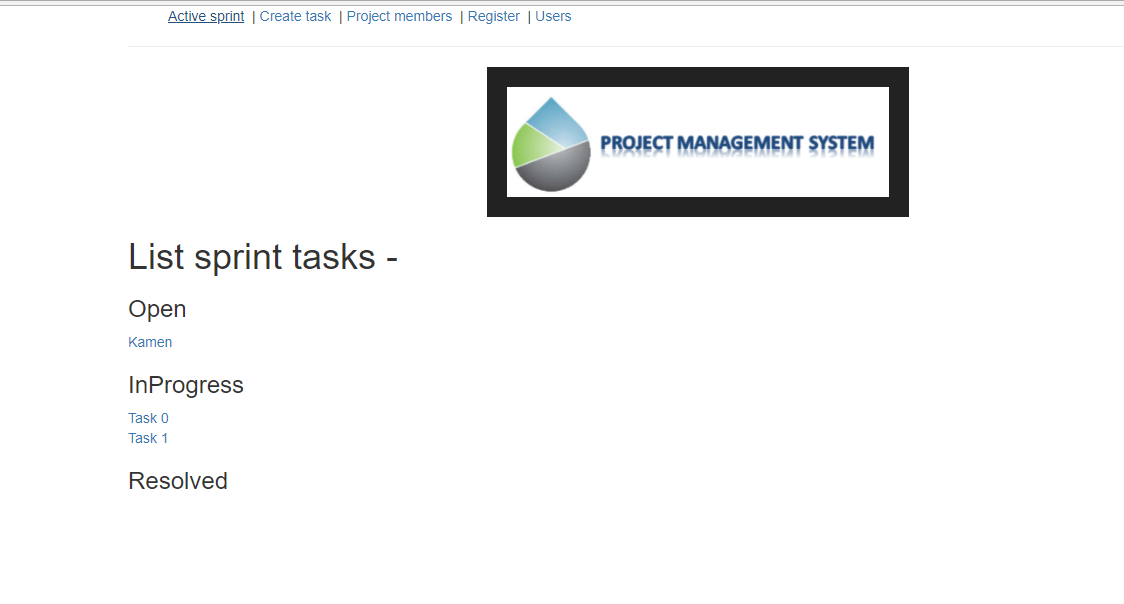
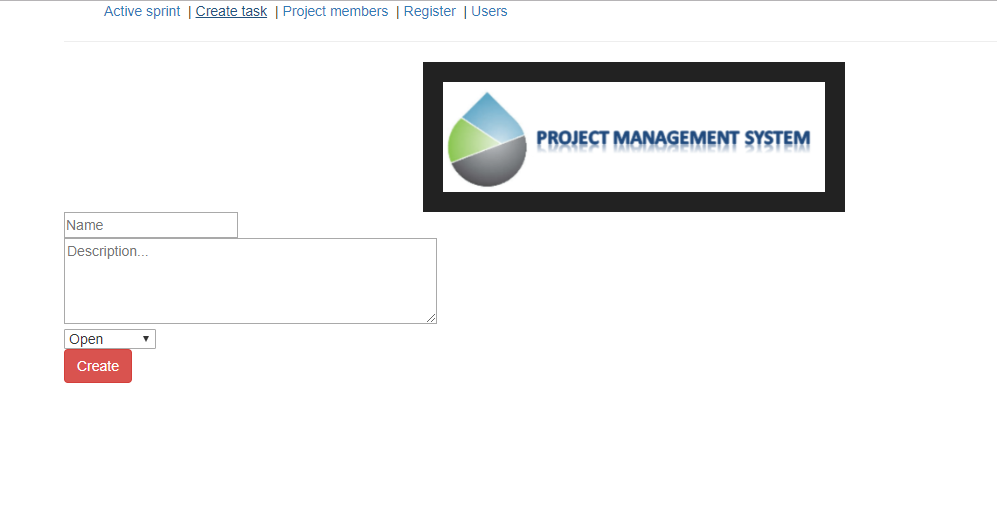
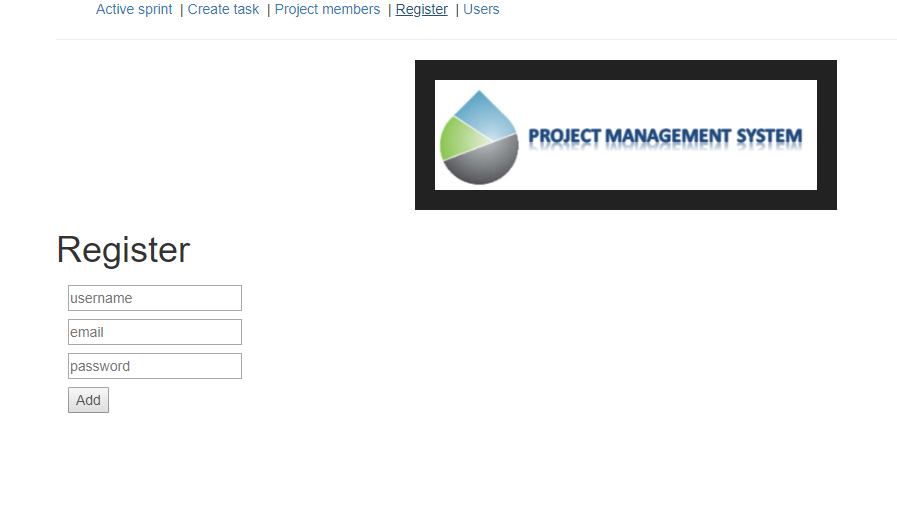
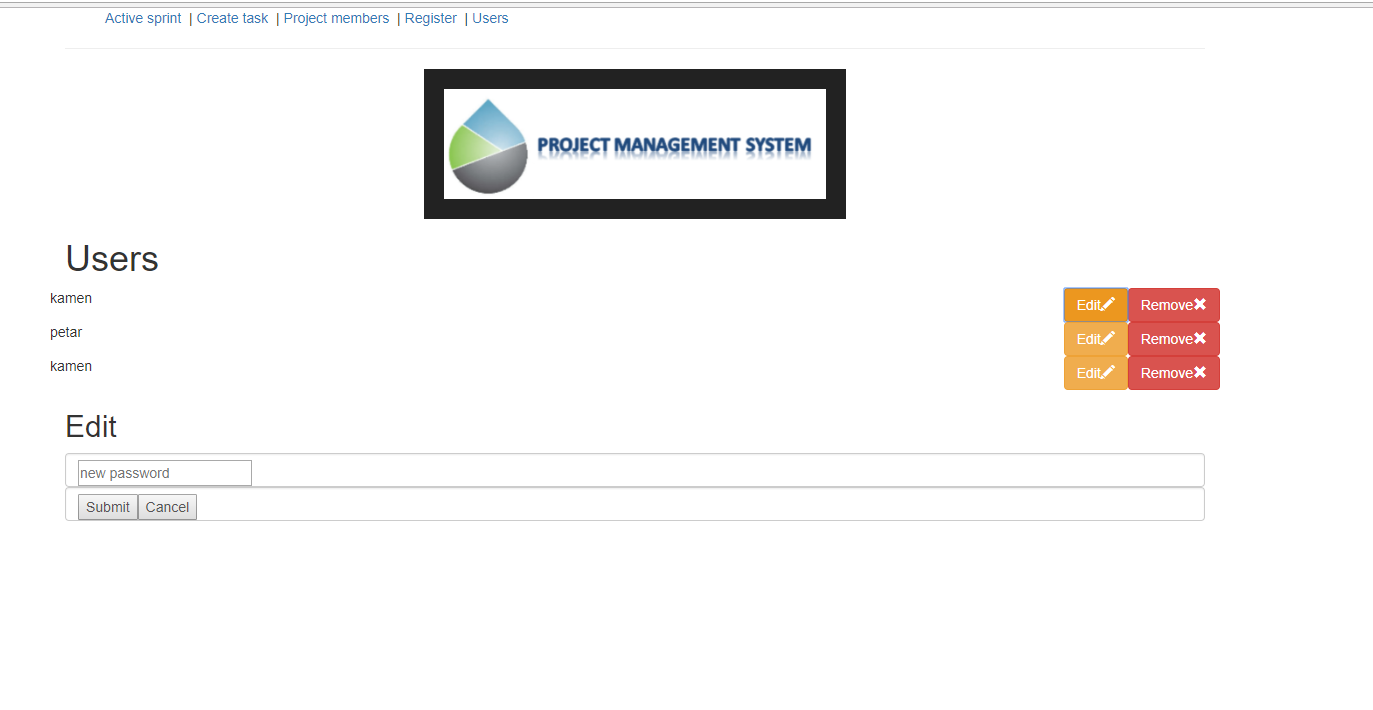
node.js v.6.10.0 or above

Проектът е наличен на адрес <https://github.com/peter359/course-react> .

След като се свали, в главната директория се изпълняват командите npm install и npm start за стартиране на сървъра.

В директирия client се изпълняват командите npm install и npm start за стартиране на клиента.

## Потребителска документация

* 1. Начална страница  
     
  2. Разглеждане на задачи в активен спринт  
     
     1. Open – задачи които все още са отворени или преотворени
     2. InProgress – задачи които се обработват в момента
     3. Resolved – приключени или затворени задачи
  3. Създаване на задача  
     
  4. Менажиране на участниците в проекта  
     
     1. Има възможност за добавяне на нов потребител като се селектира от дропдауна и натискане на бутона Add. За да се премахне от листа е необходимо да се натисне Remove
  5. Регистриране на потребител  
     
  6. Редактиране на потребителски акаунти (смяна на парола)  
     

## Заключение

Основните трудности за проекта бяха твърде големия набор от необходими функционалности за да заработи системата. В бъдеще очакваме проекта да бъде довършен и може да бъде използван за малки проекти.

## Използвани материали

* 1. <https://www.npmjs.com/package/indicative>
  2. <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
  3. <https://webpack.js.org/>
  4. <https://babeljs.io/docs/en>
  5. <https://github.com/axios/axios>
  6. <https://github.com/facebook/create-react-app>
  7. <https://expressjs.com/en/starter/generator.html>
  8. <https://www.npmjs.com/package/sqlite3>
  9. <https://reactjs.org/docs/hello-world.html>
  10. <https://github.com/iproduct/course-node-express-react>