|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2021 | |
|  | | | |
| Project author | | | |
| F№ | Name | | Email |
| 45458 | Martin Dragnev | | mddragnev@gmail.com |

|  |  |
| --- | --- |
| Project name | Internet Book Database |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Even though movies and videos have shifted books from people daily routines, there are still people who rather read a good book than watch a movie. The ***Internet Book Database (IBD)*** provides ability for registered users to contribute information to the site, rate a book, leave a review and add a specific book to ***Want to Read*** list. In addition to that users can write comments and opinions about any book. The system will be developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Node.js + express*** as backend technologies. Each view will have a distinct URL, and the routing between pages will be done client side using ***React Router***. The backend will be implemented as a ***REST/JSON API*** using JSON data serialization. The main user roles are:  • *Anonymous User* – can only view the information pages and read review.  • *Registered user* – can add new book requests, add a review, add a book to ***Want to Read*** list.  • *Administrator* – can delete all *Registered Users*, can add a book into the database. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | The *User* can browse the information views in *IDB*, and can search for a book and read reviews. | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a username, first and last name, and choosing password. A registered user can add some information about his interests. | *Anonymous User* |
| * 1. **Manage Users** | *Administrator* can see and delete any User. | *Administrator* |
| * 1. **Manage Want to Read list** | *A User can add books to read in a* ***Want to Read*** *list and remove a book from the list.* | *Registered user* |
| * 1. **Add a book** | *Registered users and Administrator can add a book in the database.* | *Registered users,*  *Administrator* |
| * 1. **Add a comment** | *Registered user can write a comments based on their understanding of specific book.* | *Registered user* |
| * 1. **Vote for rating** | Registered users can rate a specific book for a ranking. | *Registered user* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents featured today list of books and top rated books. | / |
| * 1. **Top Rated Books** | Presents all-time top rated books. | */top* |
| * 1. **Read list** | Presents all books added to ***Want to Read*** *list.* | */readlist* |
| * 1. **Specific book information** | Presents *a specific book information.* | */title* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register in *IBD*. | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **Users** | Presents the ability to manage Users  Available only for Administrators. | */users* |
| * 1. **Adding books** | Presents the ability to add books. | */addbook* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data*. Available only for *Administrators*. | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *userId*. | */api/users/{userId}* |
| * 1. **Login** | POST *User Credentials* (username and password). | */api/signup* |
| * 1. **Register** | POST a register request | */api/register* |
| * 1. **Books** | GET books and POST new *book*. | */api/books* |
| * 1. **Book** | GET, PUT, DELETE *book information* with specified *id*. | */api/books/{id}* |