## Express.js Retake Exam – SoftUni Wiki

* **Exam Rules**
* You have 4 hours - **from 9:00 to 13:00**
* When you are ready, delete the ***node\_modules***folder, make sure all dependencies are listed in the ***package.json*** file and submit your archived project at <https://judge.softuni.bg/Contests/1070/ExpressJS-17-Jun-20178-Exam>
* You are provided with **HTML & CSS** resources, which you can **modify**
* Use **Express.js** as a back-end framework
* Use **MongoDB** as a database with **mongoose**
* You can use whatever **view engine** you like (Handlebars, EJS, Pug etc.…)

## Application Overview

Get familiar with the provided **HTML & CSS** and create an application that allows users to **create** and **read** encyclopedia **articles**. An anonymous user should be able to access the main page and **view any article**, as well as **register/login**. A registered user can **edit any article**.

## Functional Requirements

The Functionality Requirements describe the functionality that the **Application** must support.

**Guests (Not Logged In)**

The **application** should provide Guest (not logged in) users with the functionality to:

* Login
* Register
* **View** **any** **article**

**Users (Logged In)**

The **application** should provide **Users** (logged in) with the functionality to:

* **View any articles**
* **Create new article**
* **Edit any article**
* **Delete only his own articles**
* **Logout**

## Database Models

The **Database** of the **SoftUni** **Wiki** application needs to support **2 models**

### User

* Username - string (required), unique
* Password - string (required)
* **Created Articles - a collection of** Articles

### Article

* Title - string (required), unique, min 5 characters
* Description - string (required), min 20 characters
* **Article Author - a User (ObjectId)**
* **Creation Date – Date (default now)**

Implement the models with the **correct datatypes**

## Application Pages

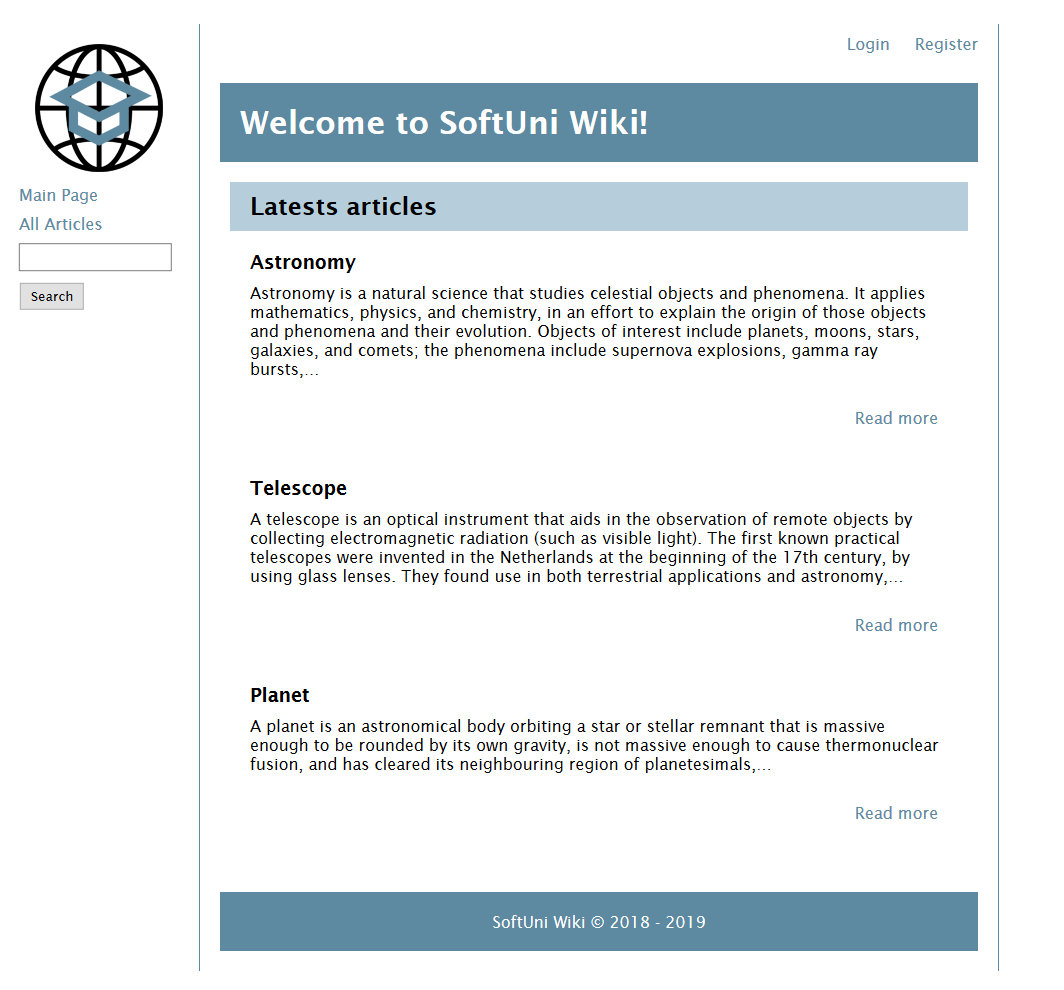
### Guest Pages

These are the pages and functionalities, accessible by **Guest (logged out users)**

**Main Page (Logged Out User)**

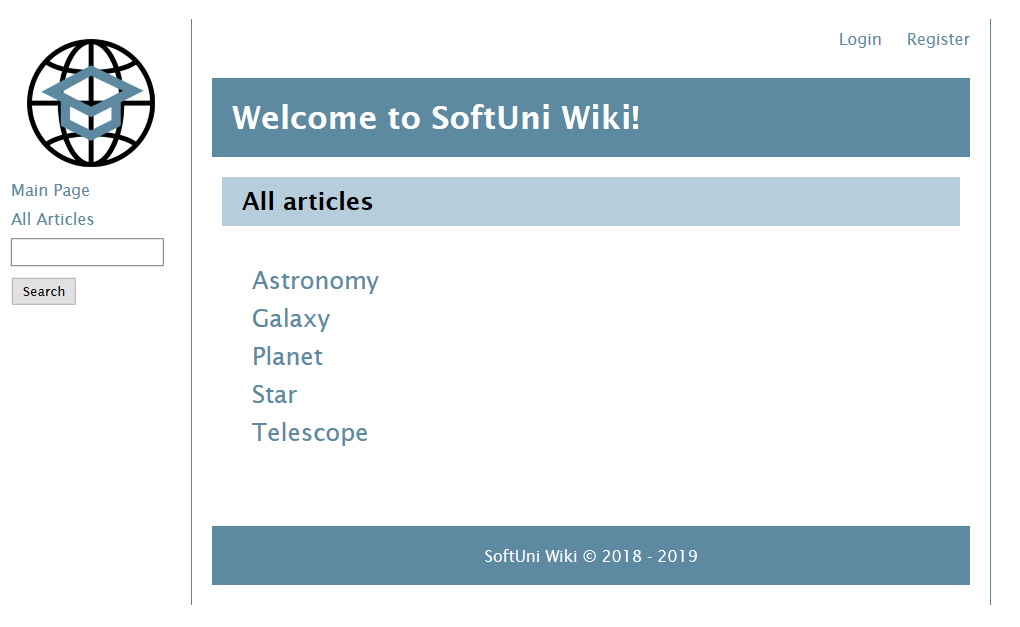
The **main** **page** should display all **navigation** menus, and the **latest 3 articles** (first 50 **words** of the content) Guest users have the option to login or register (see provided HTML or screenshots below).

Make sure only the **necessary** elements are displayed in the navigation menu and on every page, depending on the status of the user (logged in or not).



**All Articles (Logged Out User)**

Show a list of **all articles**, with a **links** to display their contents (**details article page**)



**Login (Logged Out User)**

With a **valid** **email** and **password**, a guest user should be **logged** **in**.



**Register (Logged Out User)**

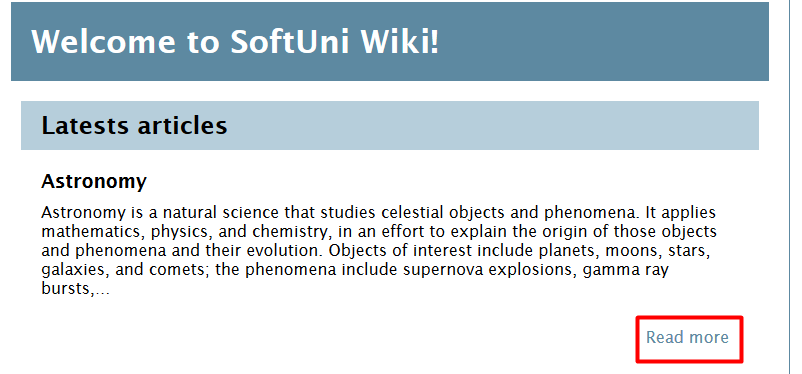
Register a user inside the database with **email** and **password**. Both **passwords** must **match**!



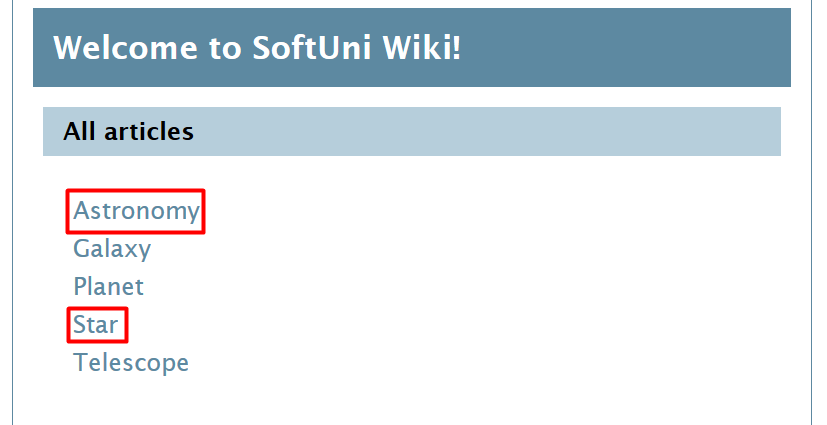
**Article Details (Logged Out User)**

**Details** article can be accessed by:

* **[Read more]** **button**

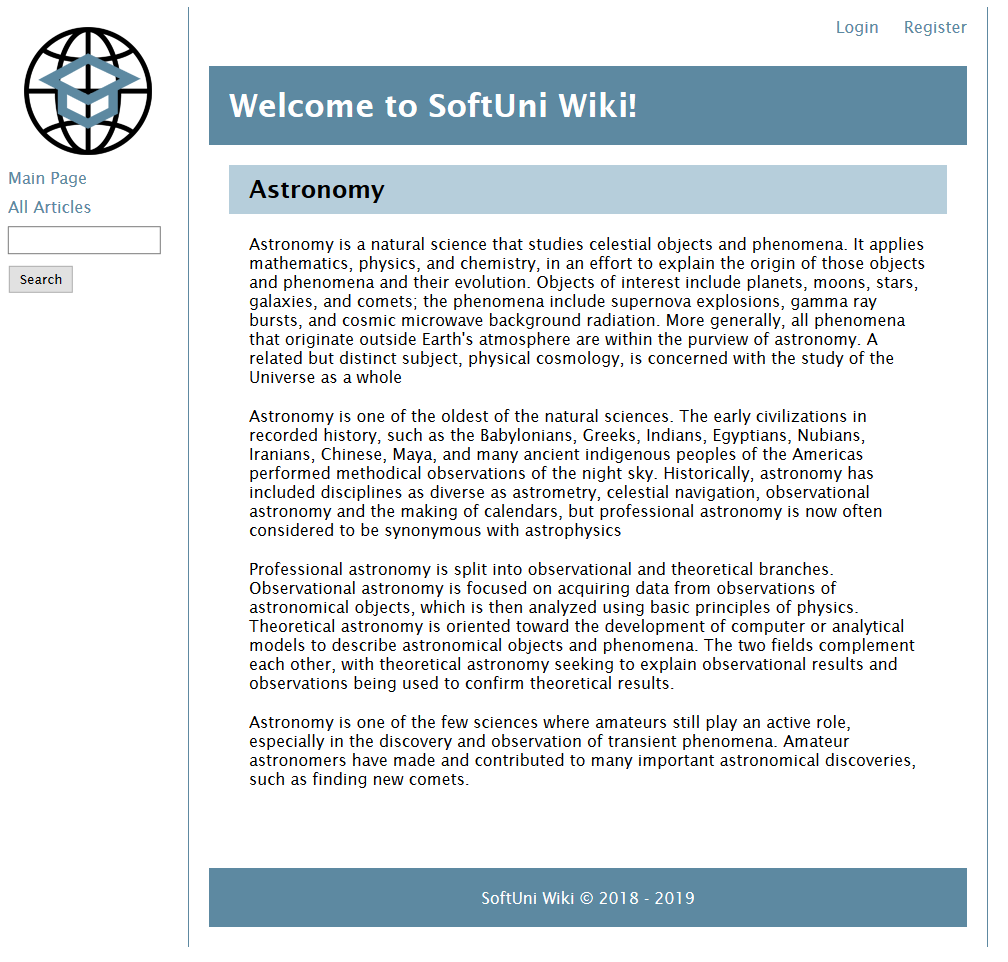


* Directly from **all articles**



Load the article from the database and display a page, containing its **title** and **content.**

**Hint**: For better looking formatting when outputting contents, interpret two-line breaks as a new paragraph.

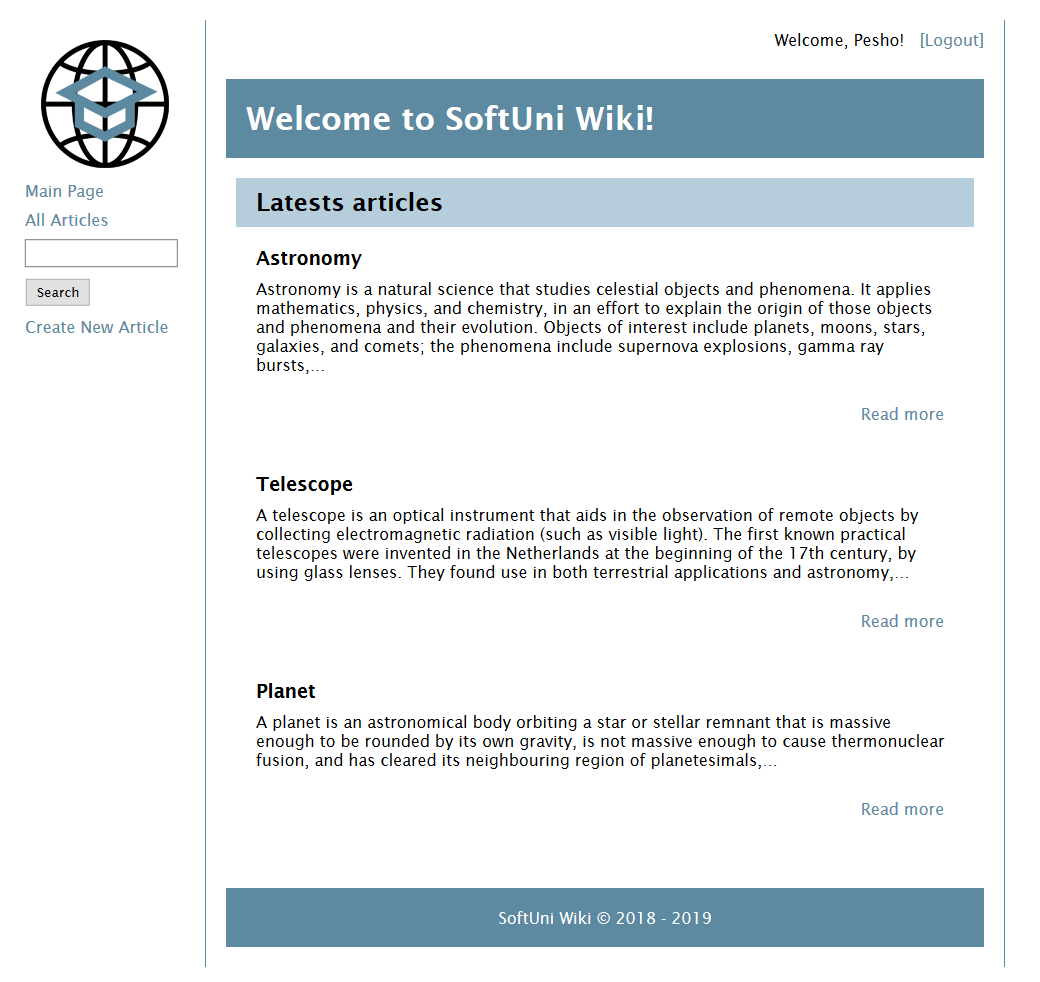


### User Pages

**Main Page (Logged in User)**

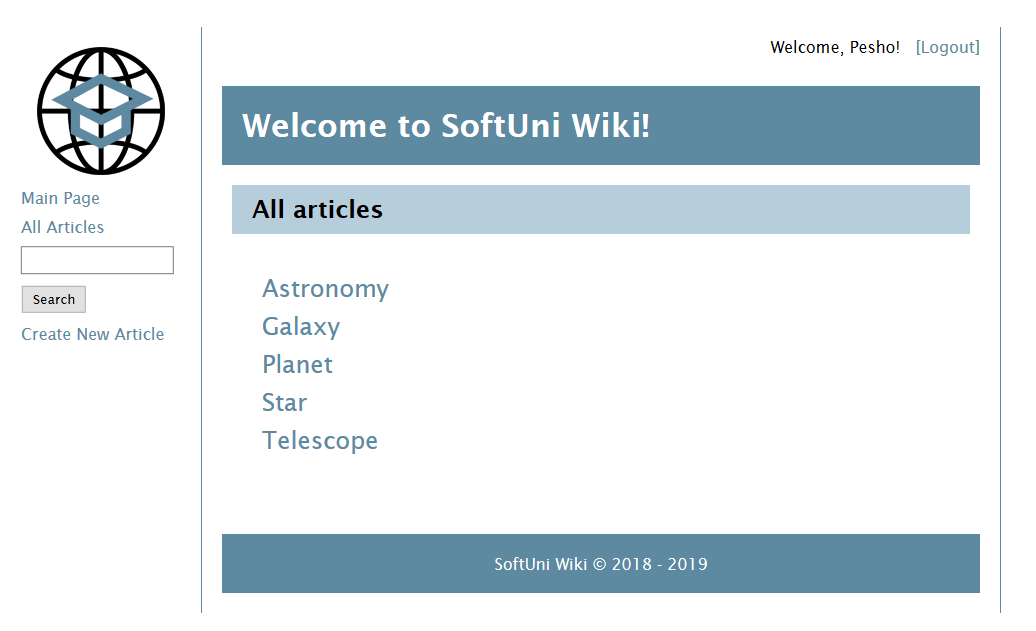
The **main** **page** should display all **navigation** menus, and the **latest 3 articles** (first 50 **words** of the content) Registered users see a greeting with option to log out (see provided HTML or screenshots below).

Make sure only the **necessary** elements are displayed in the navigation menu and on every page, depending on the status of the user (logged in or not).



**All Articles (Logged in User)**

Show a list of all articles, with a links to display their contents (details article page).



**Create New Article (Logged in User)**

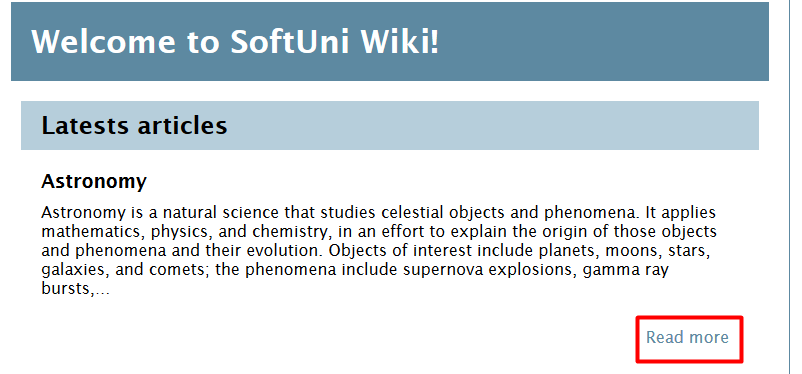
With valid **title** and **content**, a new article should be created in the database



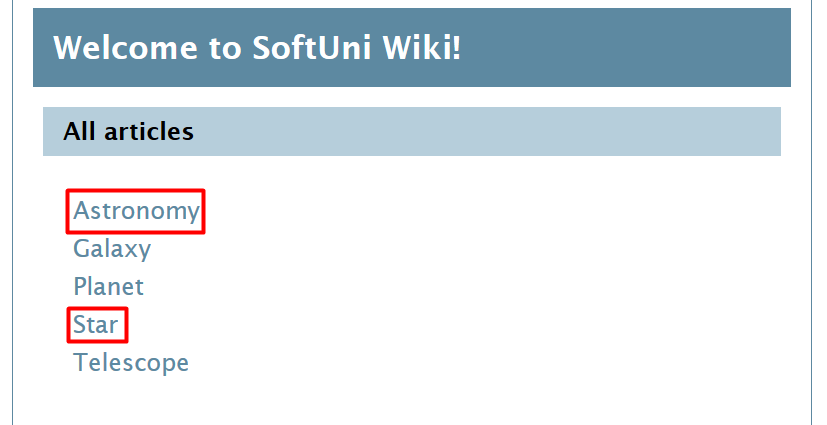
**Article Details (Logged in User)**

**Details** article can be accessed by:

* **[Read more]** **button**



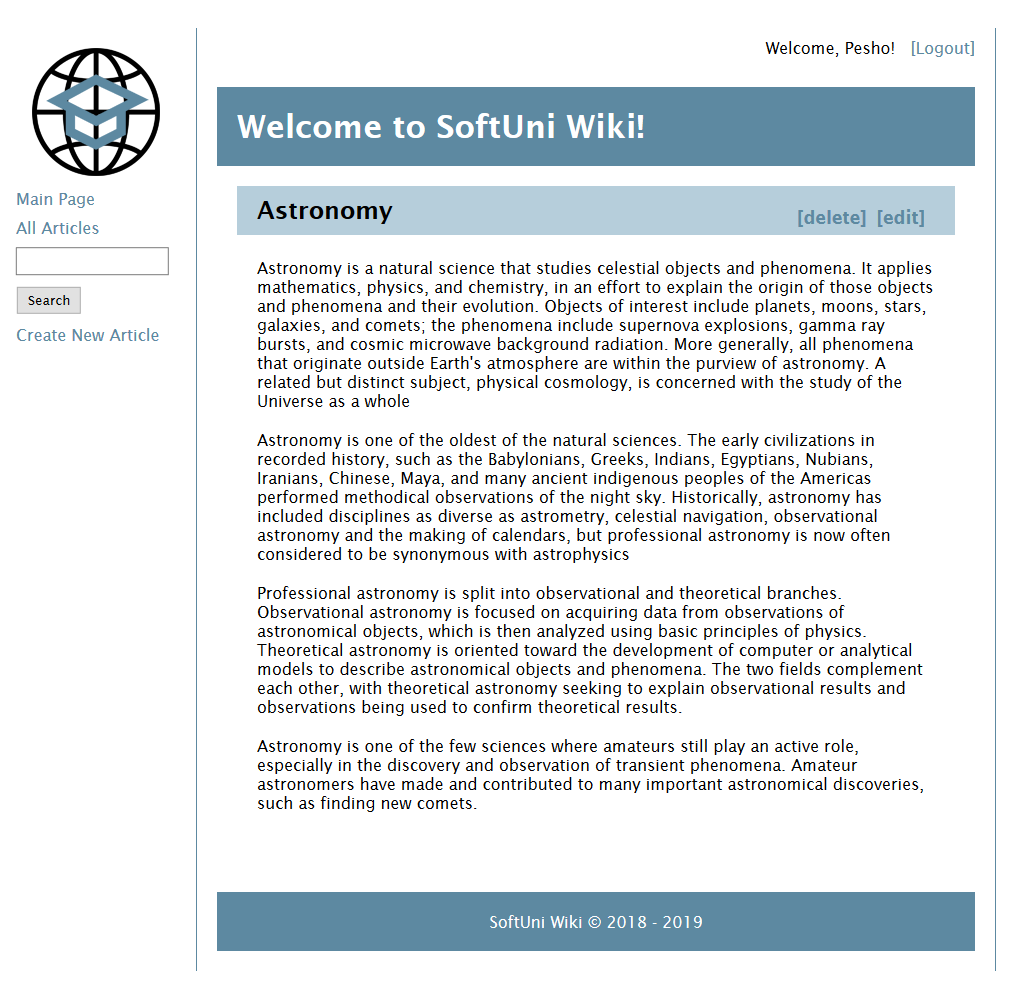
* Directly from **all articles**



Load the article from the database and display a page, containing its **title** and **content.**

**Hint**: For better looking formatting when outputting contents, interpret two-line breaks as a new paragraph.

**[Edit] button** should be shown. **[Delete] button** should be shown only if the currently logged in user is the author of that article.



##### Edit Article

An article can be edited by any **logged** **in** **user**. Only the article's description can be changed.



##### Delete Article

Only article author should be seen the **[Delete] button**. Here you have **no** **view**, just **delete** the current article from the database and to the **redirect** **main** **page**.

## Security Requirements

The Security Requirements are mainly **access** requirements. Configurations about which users can access specific functionalities and pages.

* Guest (not logged in) users can access Main page and functionality
* Guest (not logged in) users can access Login page and functionality
* Guest (not logged in) users can access Register page and functionality
* Guest (not logged in) users can access All Articles page and functionality
* Guest (not logged in) users can access Details Article page
* Users (logged in) can access **Main** page and functionality
* Users (logged in) users can access All Articles page and functionality
* Users (logged in) can access **Create Course** page and functionality
* Users (logged in) can access Logout functionality.
* Users (not logged in) users can access Details Article page and functionality
* **Users** (logged in) can **Edit** article
* **Only Article Author** (article creator) can **Delete** the current article
* **All Users** (logged in or not) can **Search** article

## Validation and Error Handling

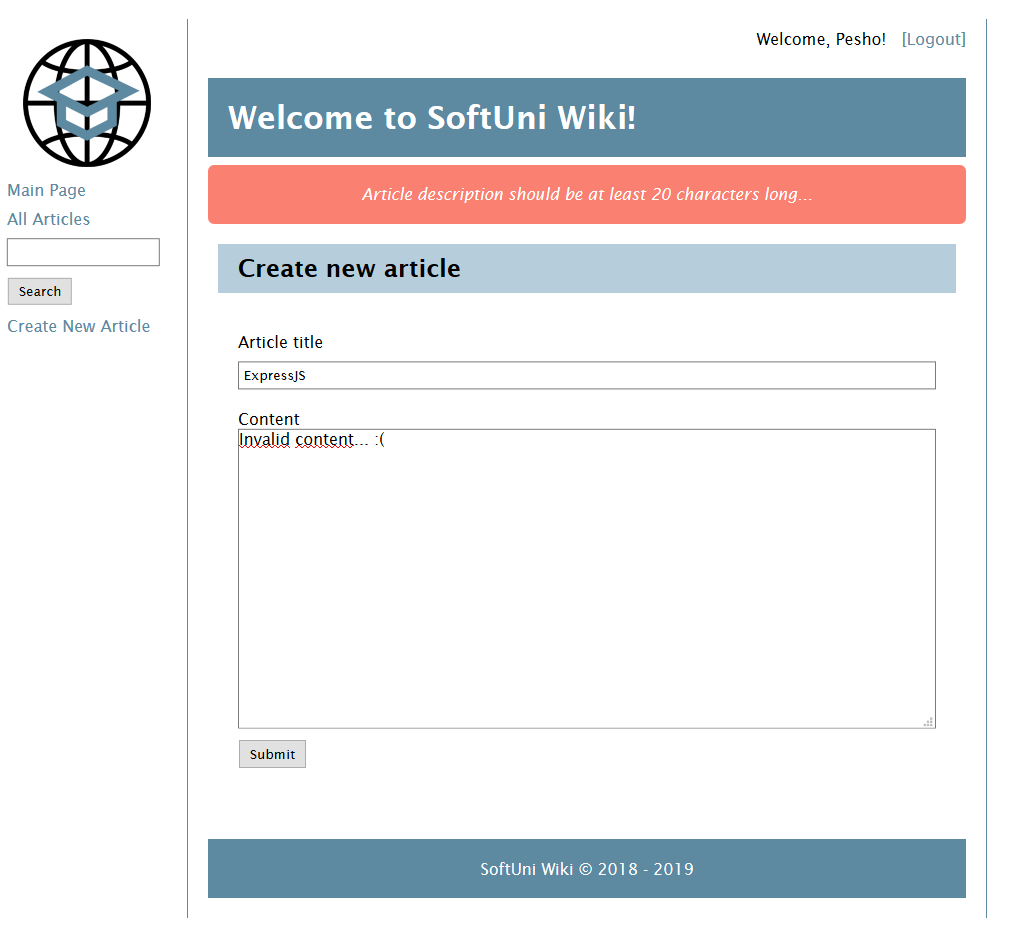
The application should notify the users about result of their actions.

### Login/Register



### Create New Article

An article must have a **title**, **description**, **author** and **editor's collection**.



## \*Bonus

When a user enters a search criteria, look in the database for **partial matches** and return a list of links to all articles, whose **title contains the input**.

