# ASP.NET Web Forms – Practical Exam – September 2013

## Library System in ASP.NET Web Forms

You are assigned to design and implement a **Library System** where visitors (without authentication) should be able to **view categories** and **books**, as well as to **register** and **login** in the system. Registered users (after login) should be able to edit the categories and books. The system should be implemented as server-side Web application in ASP.NET Web Forms.

### Library System Data Layer

Design a simple data layer to hold **users**, **categories** and **books**. Each **user** has **username** and **password**. The password should be stored in the DB encrypted (not as clear text). **Categories** have **name** (mandatory) and hold a **set of books**. Each **book** has mandatory **title** and **author** (authors) and optionally **ISBN**, **web site** (URL) and **description**.

Use **Entity Framework** as ORM engine and **MS SQL Server Local DB** as database storage engine. Your project should run after "copy/paste" deployment, without changing connection strings or other settings. You may use code first, model first or database first approach to access your data from Entity Framework.

Fill some sample data in the DB to simplify any further testing.

Use the ASP.NET Identity system to keep the users and their encrypted passwords.

10 score

### Library System ASP.NET Web Forms Application

Design and implement the Library System as a server-side web application in ASP.NET Web Forms. You may use the following steps during your work:

* **Master Page** – design an ASP.NET Master Pages to reuse the common page elements like headers and footers in all other pages in the project.

2 score

* **Configure the ASP.NET Identity System** to enable user management functionality (register / login / logout).

1 score

* **Register user** – by username and password the system should be able to register a new user in the system. After successful registration, the user should be redirected to the start page.

3 score

* **Login user** – by username and password the system should be able to login an existing user. After a successful login, the user should be redirected to the start page.

2 score

* **Logout** – successfully logged in user should be able to logout from the system. After a successful logout the start page should be shown.

2 score

* **Display the categories and books** – at the application start page display all categories and all books for each category (show their title and author).

5 score

* **View book details** – clicking on a book from the start page should display the book details (title, author, ISBN, web site and description) in a separate page.

5 score

* **Search** – at the start page a search box should allow searching by keyword / phrase. It opens a separate page for the book search results and shows all matching books. Searching displays all books having the search phrase as substring in its title or author, sorted alphabetically. Searching by empty string should return all the books.

10 score

* **Create / edit / delete categories** – successfully logged in users should be able to create / edit / delete categories. The categories should be shown in sortable table with paging (use page size 5). For each category in the table there should be "edit" and "delete" buttons. The forms implementing create, edit and delete operations could be embedded in the same page below the table or in separate page. When a category is deleted all its books are deleted as well.

Ensure your UI behaves correctly when the users enter invalid data, e.g. too long text in a text field or HTML special characters like "<br/>". Validate the data in your forms.

20 score

* **Create / edit / delete books** – successfully logged in users should be able to create / edit / delete books. The books should be shown in sortable table with paging (use page size 5). For each book in the table there should be "edit" and "delete" buttons. The forms implementing create, edit and delete operations could be embedded in the same page below the table or in separate page.

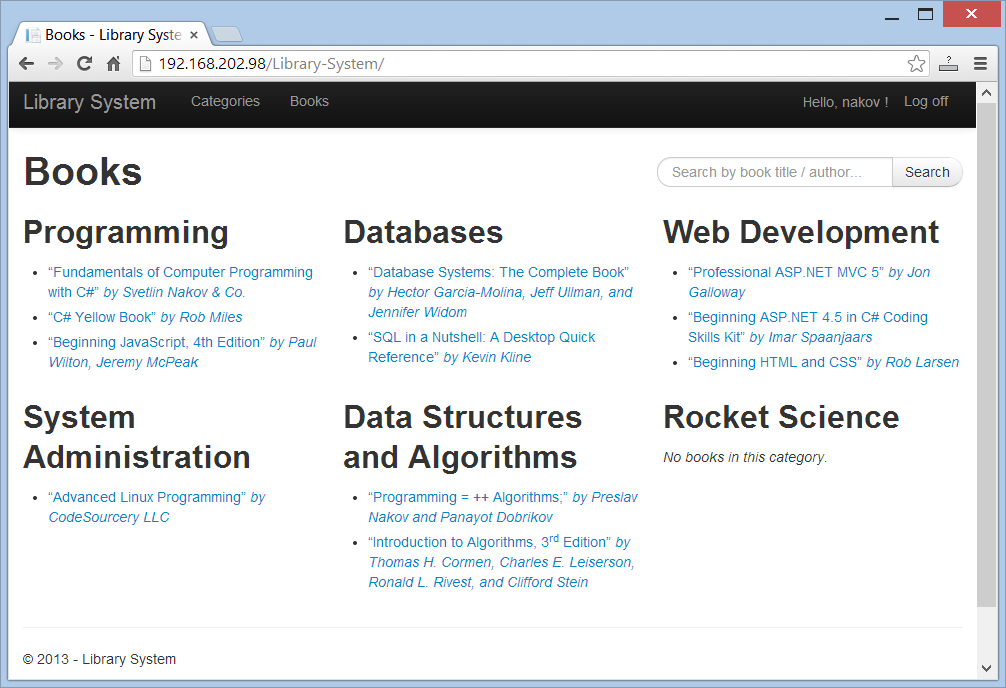
Ensure your UI behaves correctly when the users enter invalid data, e.g. too long text in a text field or HTML special characters like "<br/>". Validate the data in your forms.

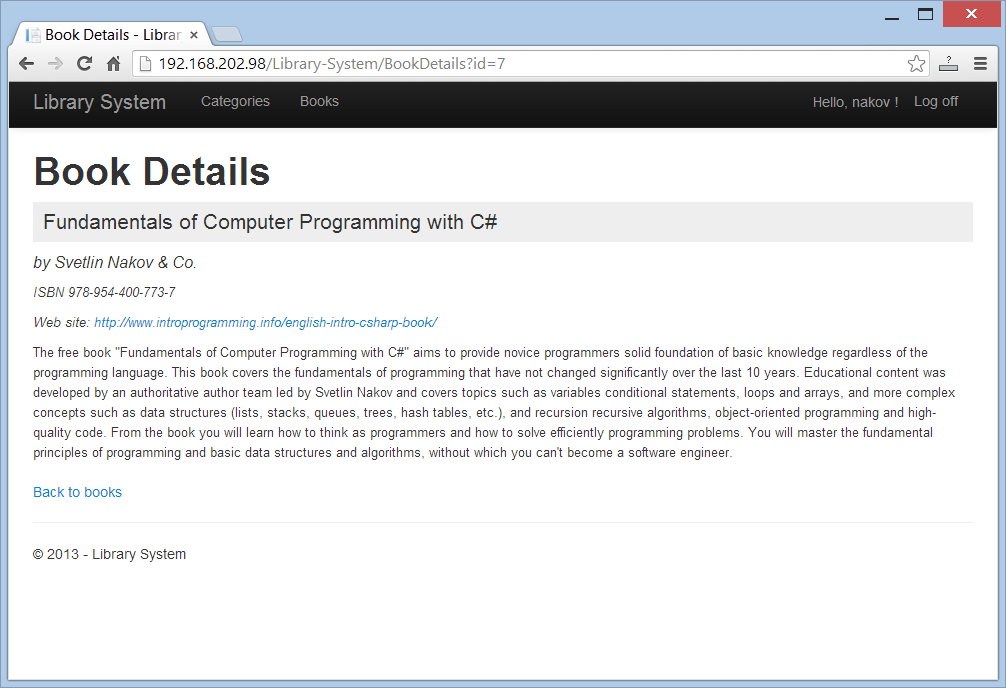
30 score

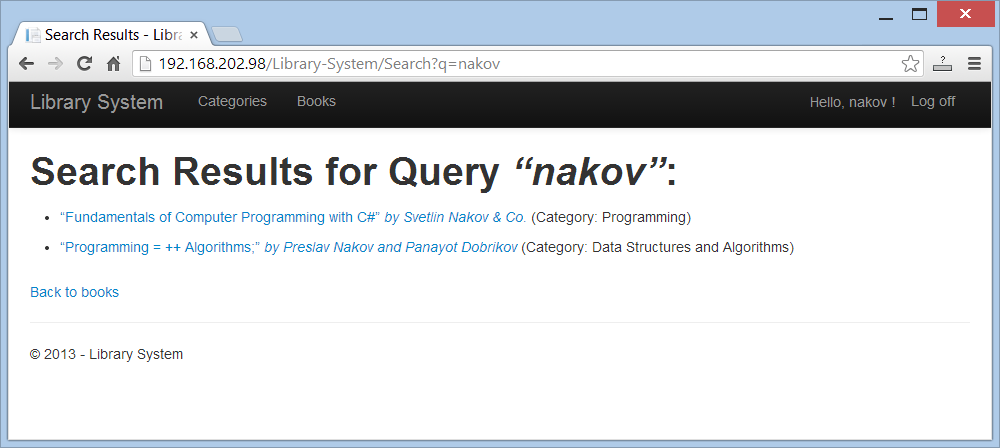
* **Error handling** – in case of error (e.g. Internet connection lost, DB connection lost, incorrect request, etc.), an appropriate error message should be displayed. You are free to decide how exactly.

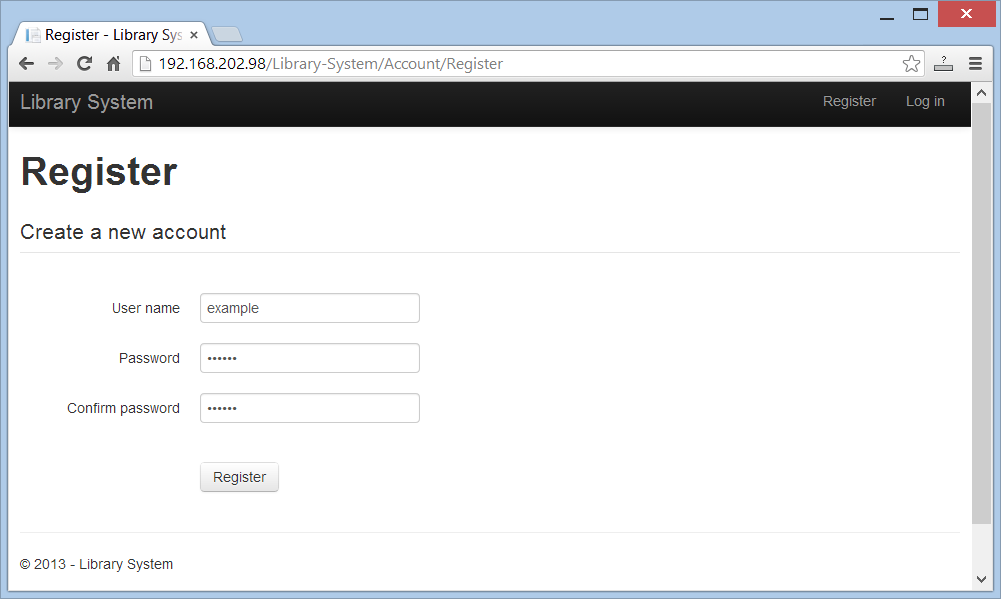
10 score

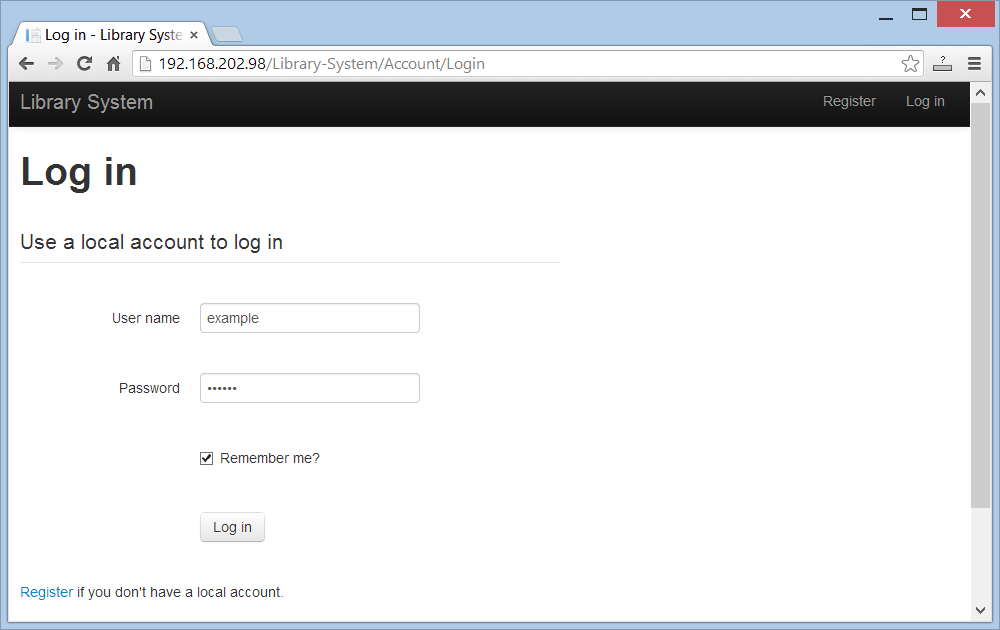
## Screenshots of the UI

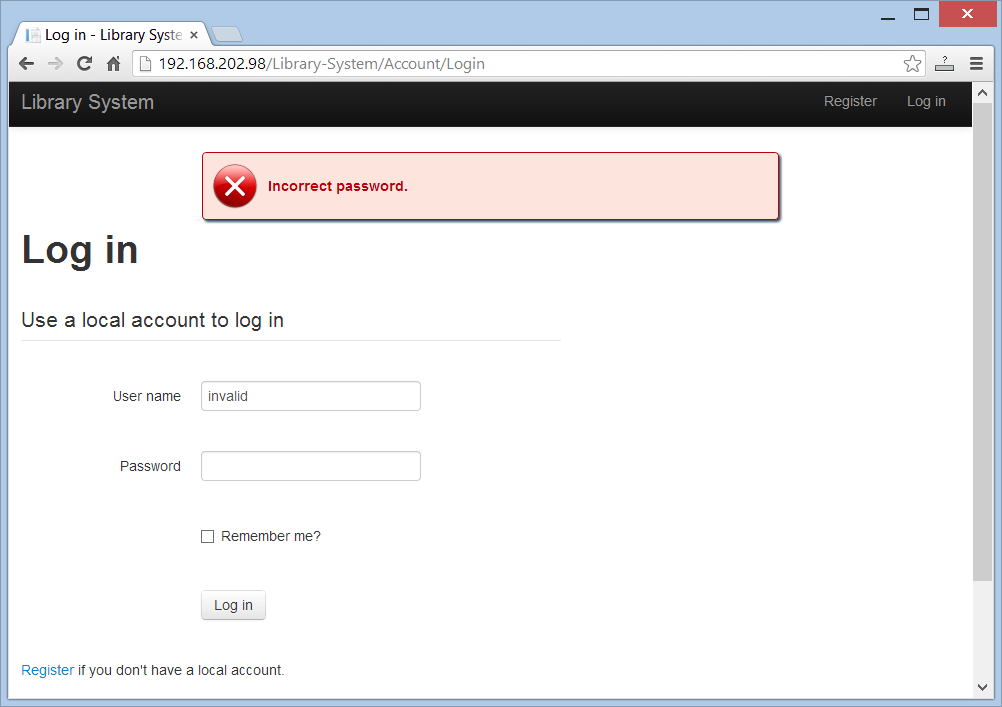


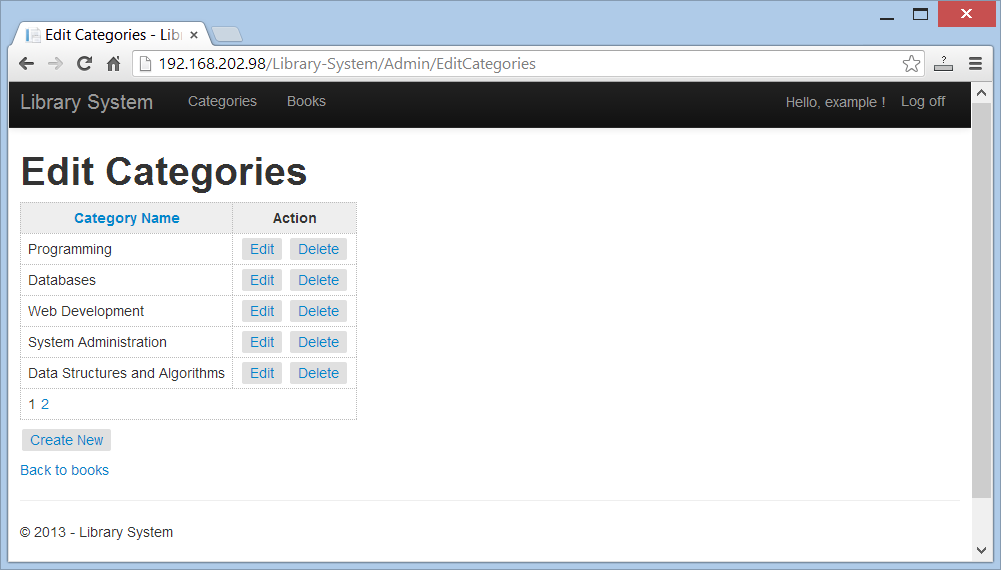


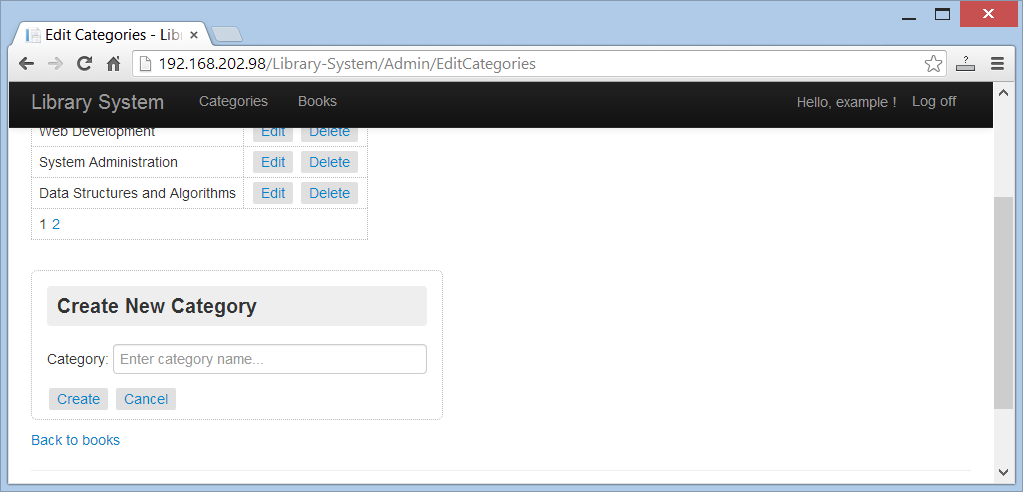


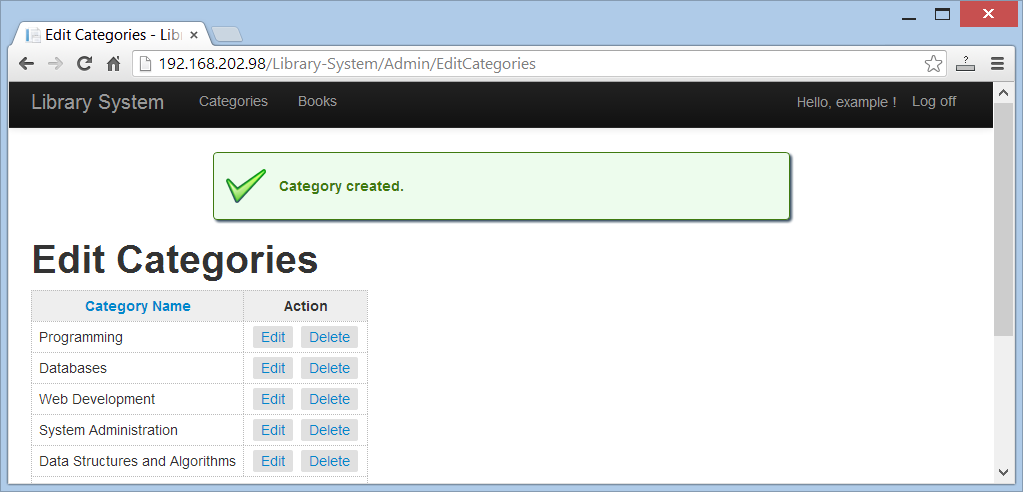


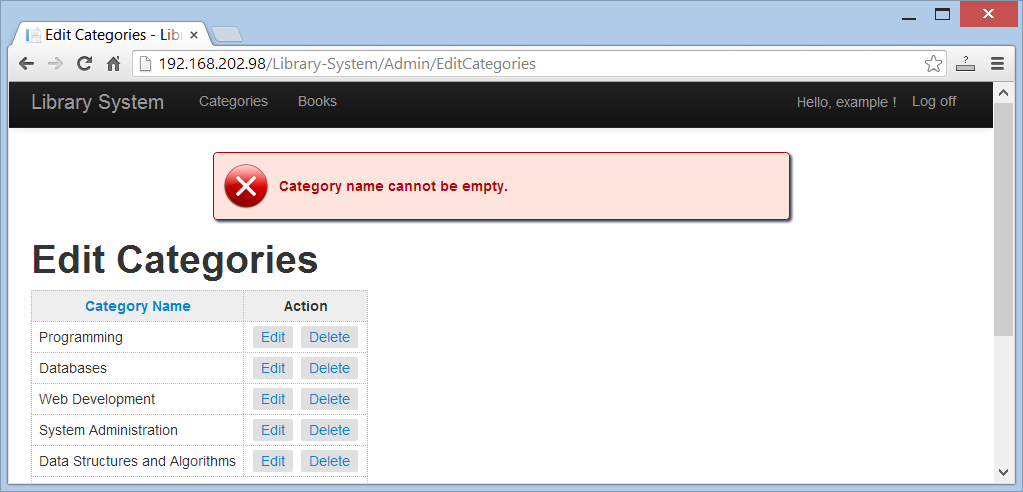


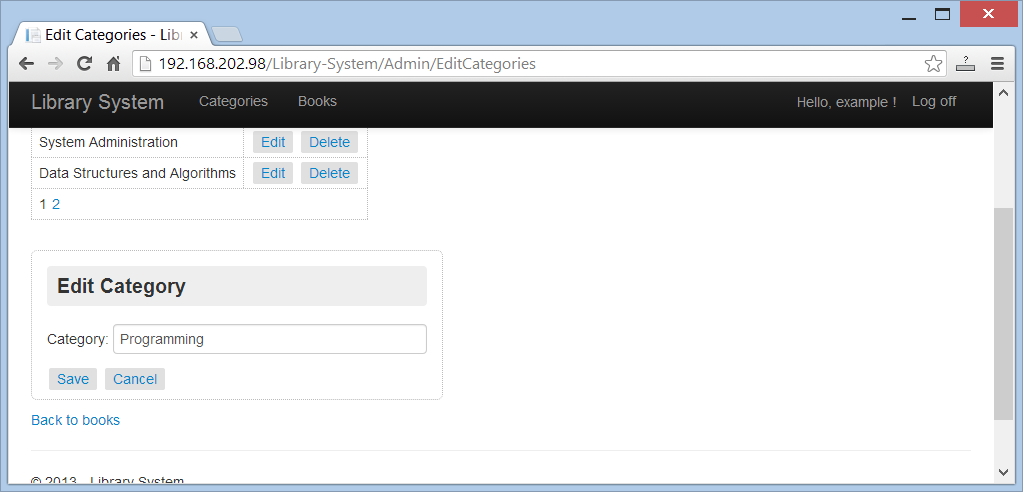


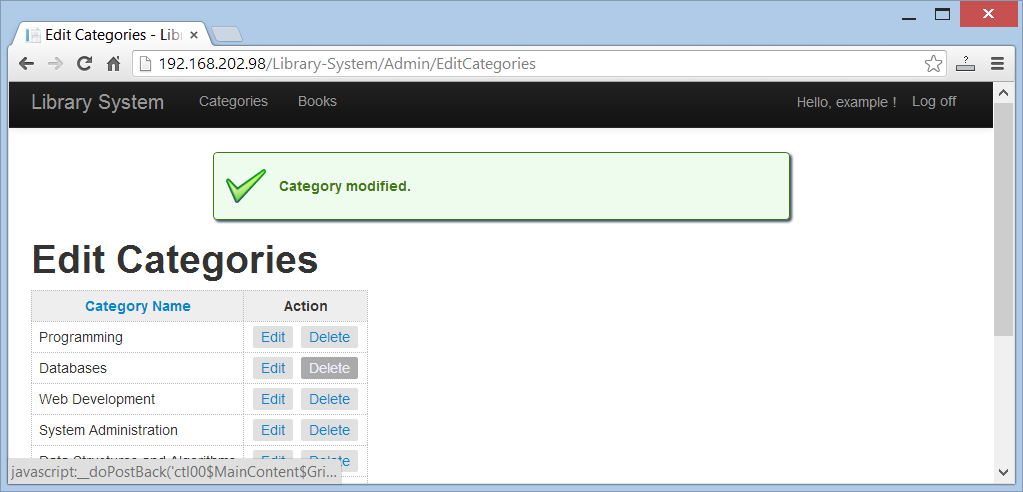


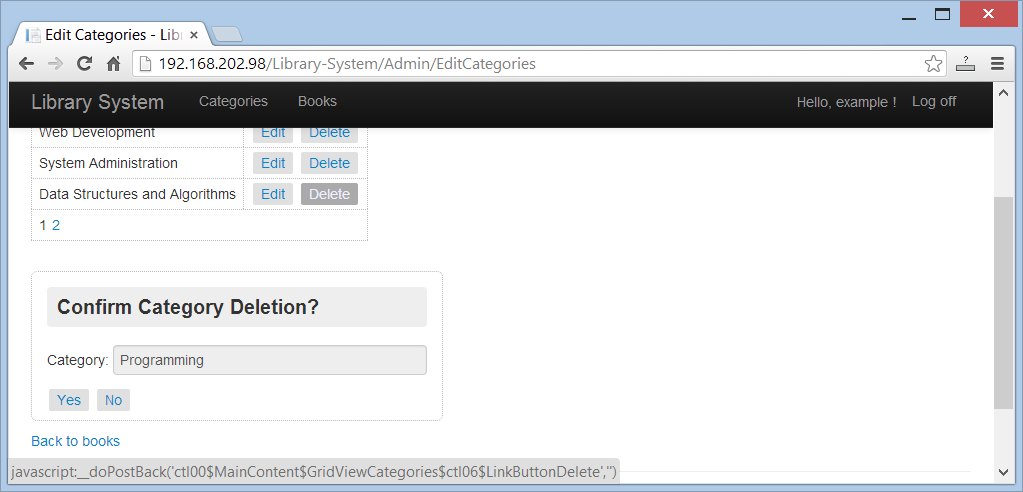


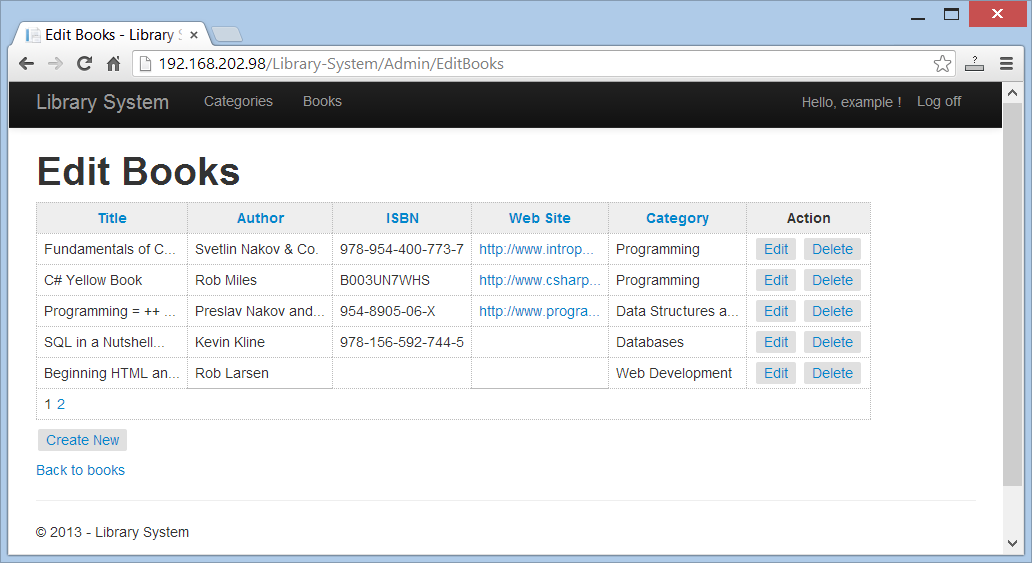


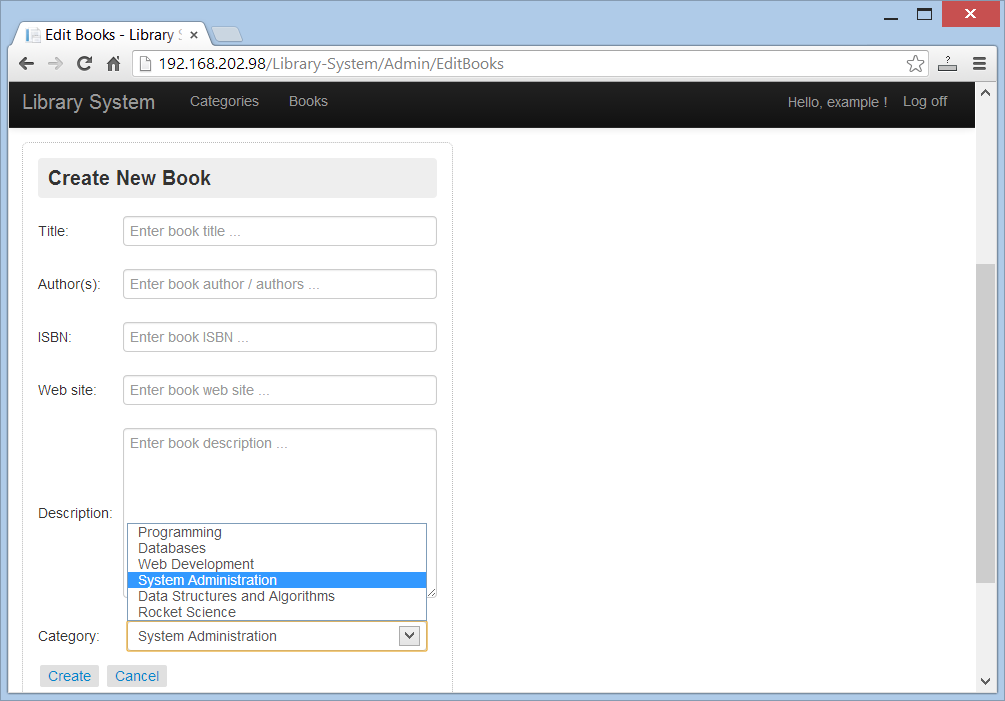


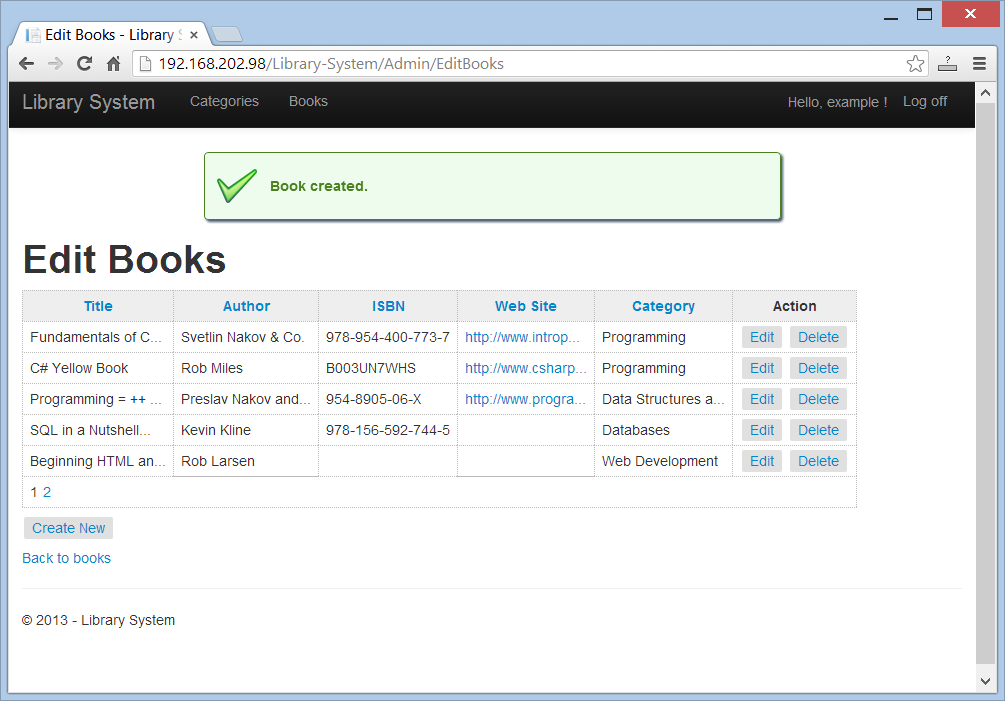


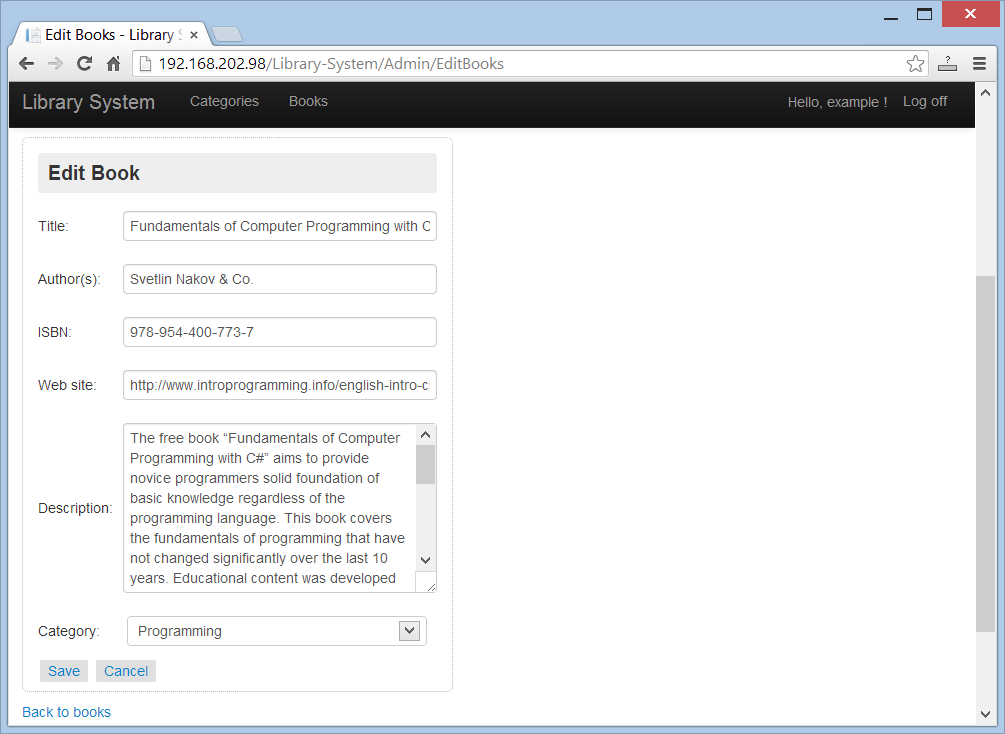


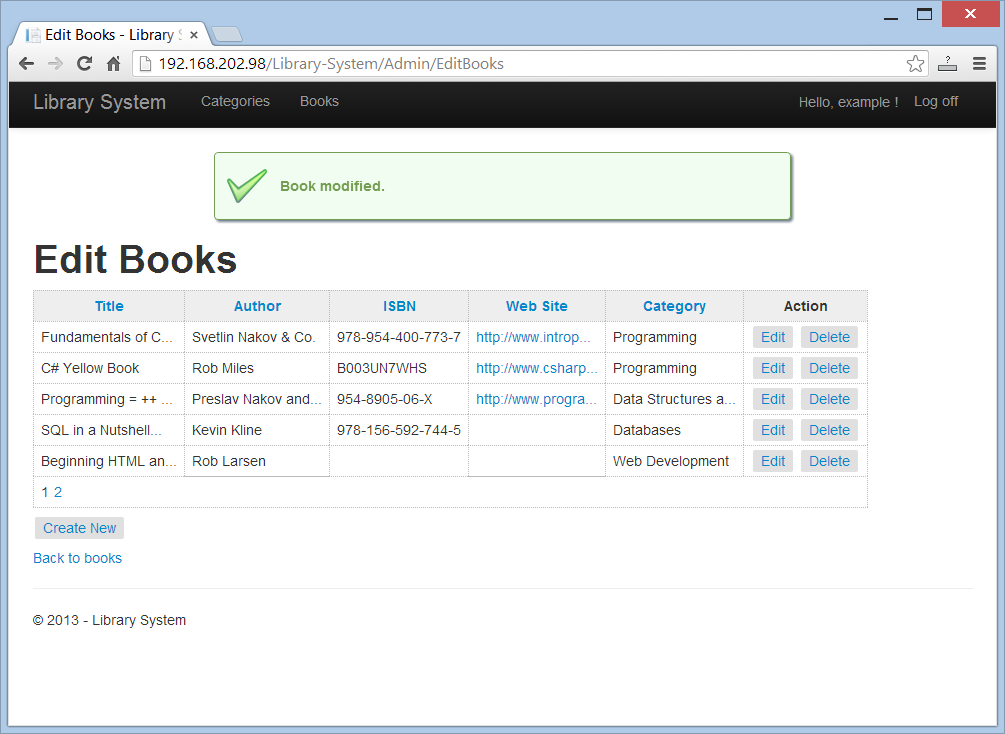


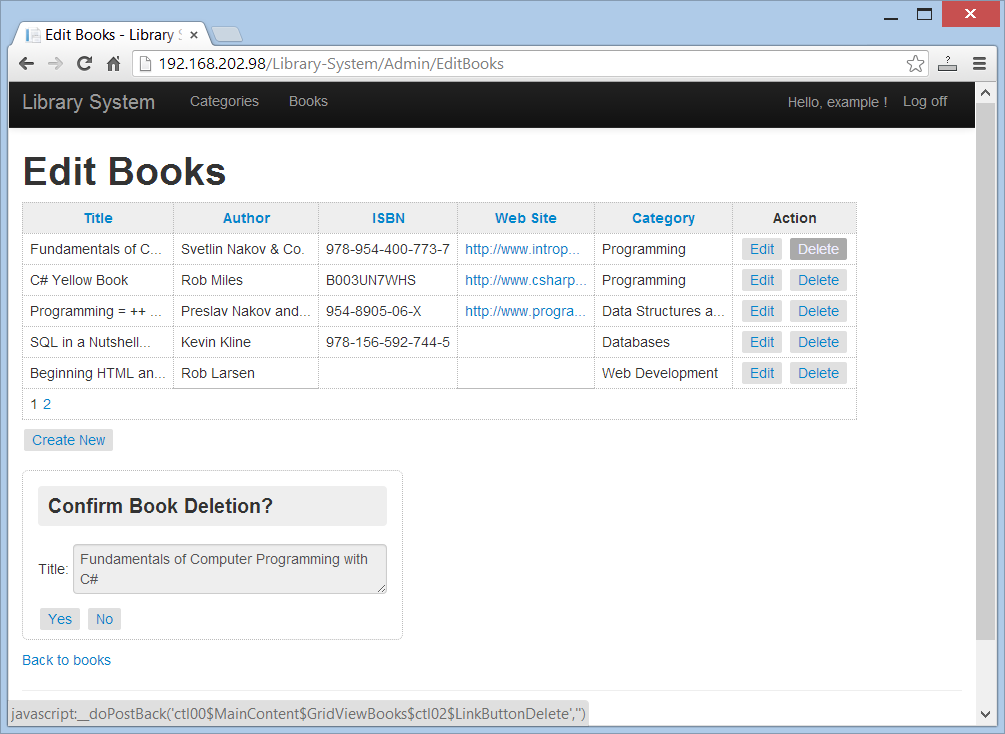


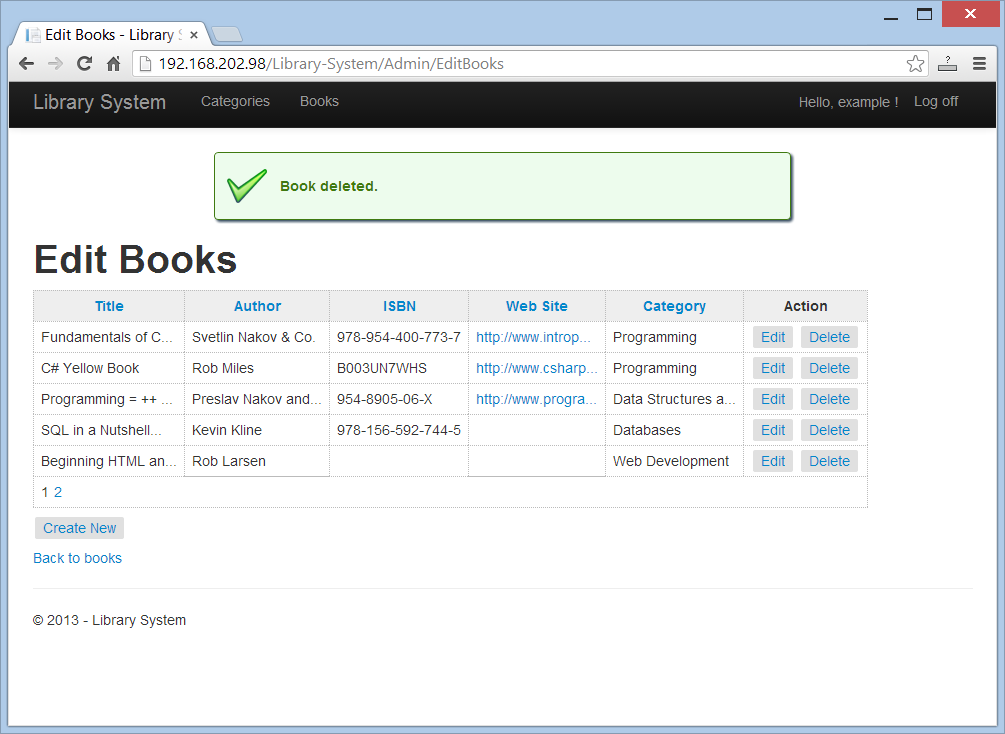












## Evaluation Criteria

The evaluation criteria include: correct and complete fulfillment of the requirements; good technical design and appropriate use of technologies; high-quality code (correctness, readability, maintainability).

To pass the exam you need to gain at least **70 score** (out of 100 score total).

## Other Terms

During the exam you are allowed to use any teaching materials, lectures, books, existing source code, and other paper or Internet resources. Direct or indirect communication with anybody in class or outside is forbidden. This includes but does not limit to technical conversations with other students, using mobile phones, chat software (Skype, ICQ, etc.), email, forum posts, etc.

## Exam Duration

Students are allowed to work up to **8 hours**.