Telerik Academy

telelink

Twitter Backup

Software System Requirements
by Telelink

1 Project description

Telelink requires an ASP.NET Core Web Application, where the users can store the posts of their favorite Twitter users and then to re-tweet some of the tweets. Design and implement that ASP.NET MVC application. The application should have:

- Public part (accessible without authentication)
- Private part (available for registered users)
- Administrative part (available for administrators only)

2 Public part

The public part of your projects should be visible **without authentication**.

This section **must** support the following functionality:

2.1 Landing page

The landing page must contain product description and documentation, team, etc. along with corresponding functionality for user register and user login.

2.2 Register page

A register page must have at least a username field, password field and email field, which are both client-side and server-side validated. Two users with the same username cannot exist.

2.3 Login page

A login page must have either username or email field.

3 Private part

Registered users should have private part in the web application accessible **after successful login**.

This section **must** support the following functionality:

3.1 Favorite Twitter accounts

User must be able to view and edit a list of their favorite Twitter accounts. That list must display some information about each account, such as followers, tweets, bio, etc.

3.2 Store tweets

User must be able to download tweets from their favorite Twitter accounts and store them inside the application. Each downloaded tweet must contain some of its metadata, such as text, tags, etc.

3.3 Review tweets

User must be able to see the text of each downloaded tweet. User must also be able to read a tweet's metadata or delete the tweet from the application.

This section **should** optionally support the following functionality:

3.4 Retweet stored tweets

User should be able to retweet their stored tweets from their Twitter account.

4 Administrative part

System administrators should have administrative access to the system and permissions to administer information objects in the system.

This section **must** support the following functionality:

4.1 User management

Admin must be able to update and delete existing users, their stored tweets and their favorite Twitter accounts.

4.2 Statistics

Admin must be able to see statistics of each user's favorited Twitter accounts, stored tweets and number of retweets (if applicable). Admin must also be able to view total downloaded tweets and total retweets.

5 General requirements

Your Web application should use the following technologies, frameworks and development techniques:

This section **must** support the following functionality:

1. Use **ASP.NET Core MVC 2.0+**
2. You should use **Razor** template engine for generating the UI
3. Use **MS SQL Server** as database back-end
    a. Use **Entity Framework Core** to access your database
    b. Using **repositories and/or service layer** is a must
4. Use at least **1 areas** in your project (e.g. for administration)
5. Create responsive design
    a. You may use **Bootstrap** or **Materialize**
6. Use the standard **ASP.NET Identity System** for managing users and roles
7. Use **AJAX form** communication in some parts of your application
8. Use **caching** of data where it makes sense (e.g. starting page)
9. Apply **error handling** and **data validation** to avoid crashes when invalid data is entered (both client-side and server-side)
10. Prevent yourself from **security** holes (XSS, XSRF, Parameter Tampering, etc.)
    a. Handle correctly the **special HTML characters** and tags like <script>,
, etc.

11. Use **GitHub** and take advantage of the **branches** for writing your features.
12. Create unit tests for your business functionality following the best practices for writing unit test

This section **should** optionally support the following functionality:

1. **Documentation** of the project and project architecture (as .md file, including screenshots)
2. Setup CI/CD with Jenkins on Kestrel behind a local IIS instance
3. Login and register with a Twitter Account (Social Login)
4. Deploy in the Cloud (Azure)

6 Partner Evaluations

May 8: Students present their projects in front of Telerik Academy Trainers and Partners. The partners evaluate their respective projects. Below you'll find more information about the evaluation process and criteria.

May 9: Final defenses and evaluation of all students for their work in front of Telerik Academy trainers. No participation is required on behalf of the partners for this activity.

Partner representatives should evaluate the project according to the following criteria:

1. Feature Completeness and Quality
   - Have all requirements been met? Have optional requirements been implemented?
   - Are there any issues with performance, bugs, or malfunction?
2. Code Architecture and Design Quality
   - Loose coupling
   - Modularity
   - Extensibility
3. User Experience and Creativity

The final grade of the project is determined by the following rules.

Each of these 3 criteria are awarded points from 0-5, where 0 is the lowest score and 5 is the highest:

0	1	2	3	4	5
Missing	Poor	Fair	Good	Very Good	Excellent

Then the final project grade is determined by the following formula:

[Feature Completeness] * 3 + [Code Architecture Quality] * 2 + [User Experience and Creativity] * 1.

Maximum number of points is 5*3 + 5*2 + 5 = 30.

This final score should be awarded by partner representative on **May 8**