News Website

Plop Teodor(GMRV 2), Voicila Eduard(GMRV 2)

Project Description

The project consists of a news website, with multiple types of users:

- Anonymous (can view news)
- Normal User (can add comments to news, can propose news for publication)
- Editor (can publish news or publish normal user news)
- Administrator (can delete published news and change other user permissions)

News are split into multiple categories, which can be edited by the administrators. Viewers can sort / filter news or use the search function to find information within news.

User Interface

The website has a master page present in all its pages.

The master page has a navigation bar on the top side of the screen. This menu will contain a link to the home page, links to each of the news categories, a search feature and a Login/Logout button. Depends on which user is currently logged in, new buttons may become available, like Propose / Publish / Admin.

Below the top bar menu we have a carousel slide having images for each of the news categories. Clicking one of these images will redirect you to the chosen news category.

The rest of the UI is page dependent:

Home Page

Contains all the news, starting with the most recent ones. Each news is made of an image, a title and content. You can also see who created the article and when.

Login/Register Pages

Simple input areas for Username and Password, as well as one Sign In and one Sign Up buttons.

Propose Page for Normal User & Publish Page for Editors

Single line input area for Title, Multi line input area for content, a dropdown for the news category, the option to upload an image and the Submit button.

Propose Page for Editors

Editors can view all proposed news here. The way news are shown is similar with the *Home Page*, the only difference being two buttons on top of each proposed news: Delete and Publish.

Admin Page for Administrators

This page shows a table containing all users and an Edit button on each row.

Data Model

- Users Table
  - Username (PK)
  - Password
  - Status
- Domains Table
  - Name (PK)
- News Table
  - ID (PK)
  - Title
  - Content
  - Publisher (FK - Users Table)
  - Date
  - Domain (FK - Domains Table)
  - Image
- Users News Table (same structure as News Table, used for holding unpublished news)
- Comments Table
  - ID (PK)
  - NewsID (FK - News Table)
  - Username (FK - Users Table)
  - Text

Architecture of the project based on components

SessionManager is a singleton type instance that holds information about the current session. It is used in the app to gather information about the current logged in user: name & permissions.

Each table must inherit from TableBase, which receives a connection string and manages the connection with the database. Therefore, we have these table classes: *UsersTable, NewsTable, DomainsTable, CommentsTable, UsersNewsTable*. Each of these classes implement queries of their own (Select user by name, Select news by id, etc.). Each of these functions return a boolean value, to know if we were able to successfully retrieve the information, and an error string in case we were unable to. Each of these table classes have static functions so we can access them from anywhere we like in the pages.

Arguments are interpreted using the *Arguments* class, which takes in a request and looks for any known arguments.

Technologies used in this project

We used SQL queries for database connection, ASP.NET for backend as well as a bit of frontend, and bootstrap library for frontend.

ASP.NET sends string queries to gather data from SQL. Gathered data is then dynamically generated in the page using ASP.NET functions. For frontend, we pass in bootstrap classes.

Distribution of tasks on each team member

Tasks were split into two major groups:

- Backend (Plop Teodor)
- Frontend (Voicila Eduard)

Backend Tasks

- Create database, tables and connection between tables
- Establish connection to database
- Implement SQL queries inside ASP.NET
- Generate Menu on Master Page
- Interpret URL arguments
- Register & Authenticate
- Different user permissions
- Error messages shown on master page
- Publish news
- Show published news, delete published news
- Upload images
- News categories, filter, sort, search
- Admin panel
- Propose news, publish & delete proposed news
- Implement comments to each news

Frontend Tasks

- Integrate bootstrap & jQuery
- Change menu to navigation bar
- Integrate username, search feature, login/logout button in navigation bar
- Implement carousel slide in the master page
- Link carousel images with news categories
- Beautify Login & Register pages
- Beautify alert messages, add option to close the alerts
- Propose for Normal Users & Publish pages design
- Propose for Editors page design

- Adapt Propose for Editors design to Home page
- Collapse comments section on news
- Make admin table look decent
- Centre filter & sort feature

Project Results

We have a stable website, which can be used as a starting point for a news website where simple users can help by proposing articles, editors can keep published content in check and administrators can decide permissions given to each user, given their activity and quality of their content.

Conclusions

The project has its basic features implemented and can be served as a template for this type of website, but it still needs improvements. A few examples of improvements that might come in handy:

- Modal boxes for Login/Register
- Get rid of postback for client side requests (for convenience, they have been implemented as postback for now)
- Administrators should be able to modify or add new categories for news (this implies a way to modify carousel content)
- View activity of each user
- Ratings for each news, so administrators can rank users by rating. Users with high ratings might become editors

References

https://docs.microsoft.com/en-us/dotnet/

https://msdn.microsoft.com/en-us/library/fksx3b4f.aspx

https://support.microsoft.com/en-us/help/323246/how-to-upload-a-file-to-a-web-server-in-asp-net-by-using-visual-c-net

https://getbootstrap.com/docs/3.3/getting-started/