# Softuni Wiki

# 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 home page and **view any article**, as well as **register/login**. A registered user can **edit any article**. Your **application** should also have **administrators**. Admins can **lock and delete articles**. Locked articles can’t be edited by non-admins.

Since anyone can edit any article, it’s important that **edit history** is stored for **each article** and previous versions can be accessed in their entirety. Design your models in such a way as to allow for **multiple older versions** of an article to exist and be viewable.

# Application Deploy

Server is deployed in Heroku , server address is : <https://softuni-wiki-server.herokuapp.com/>

Web app(client) is deployed in icn.bg . address is : wiki.android.bg

GitHub of app is in : <https://github.com/kgyorev/Project-react-softuni-wiki/>

Where in GitHub is code for server and client

NOTE: if you want to run server in local host you need to change remote.js file

**const** host = **'http://localhost:1337/'**; // for local Host

**const** host = **'https://softuni-wiki-server.herokuapp.com:443/'**; // For Heroku server

## Home Page & Navigation

The **home** page should display all **navigation** menus, an excerpt of the **latest article** (first 50 **words** of the content) and links to the three most **recently added articles**. Registered users see a greeting with option to log out, while anonymous users have the option to login or register (see provided HTML).

Make sure only the **necessary** elements are displayed in the navigation menu and on every page, depending on the status of the user (anonymous, logged in, admin).



## Create Article

***For authenticated users***

An article must have a **title**, a **locked status** and a list of **edits**. Each edit has an **author**, a **creation date,** **content**, and an associated **article**. Upon creation of the article, its title is stored, the locked status is set to false and the **contents**, which the user has entered, are **stored in an edit**, associated with that article. The user becomes the author for the edit and the creation date is set to the current time.



## List All Articles

Show a list of all articles, ordered alphabetically, with a links to display their contents.

Display is with pagination.



## Display Article

Load the article from the database and display a page, containing its **title** and **content of the most recent edit**. Authenticated users see the option to edit the article.

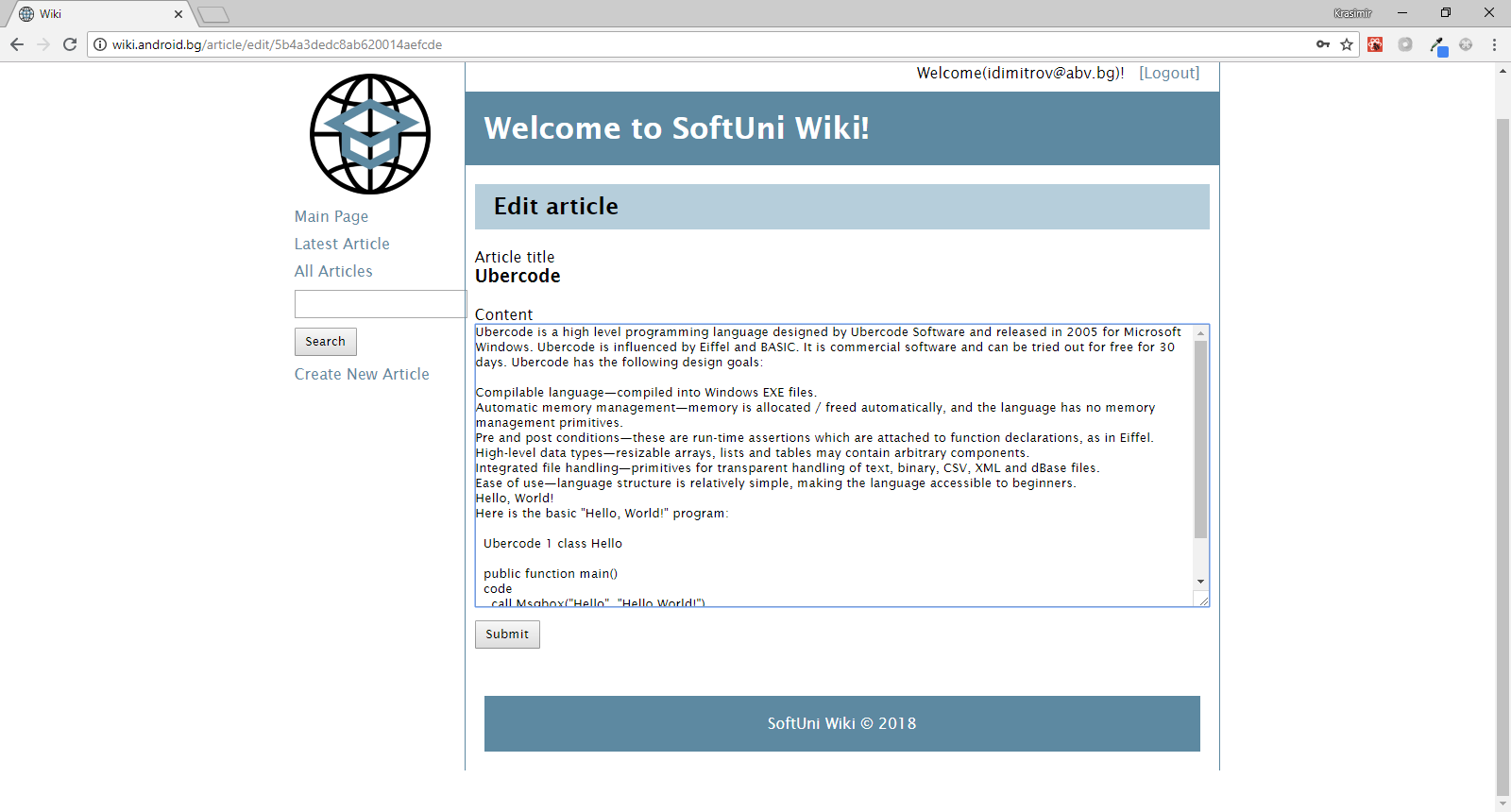


## Edit Article

***For authenticated users***

An article can be edited by any registered user. Upon submission, store the contents in an **edit** and associate it with the article. An edit has an **author**, a **creation date,** **content**, and an associated **article**.

If an article is **locked**, only admins can edit it.



## Article History

***For authenticated users***

Show a list with the **edits** for the selected article. Each edit is shown with its **creation date** and **author**. When an edit is selected, a details page with its **contents** is displayed (same as **Display Article**).



## Lock/Unlock/Delete Article

***Admin functionality***

When an article is locked, only admins can edit it. Regular users can still read it and view its history. When an article is created, it’s unlocked by default.

Admins could delete article.





## Search

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**.

Display with pagination.

