Requirements to Fairy Teller project

1. Introduction

  This project is a Web-application, the aim of which is expansion of reader community all over the world. On this site users will be able to write their own works, read masterpieces of the famous artists and amateurs, order works they like.

There won’t be any chats between users. Buying process will be fake.

2. User requirements

2.1 Program interfaces

Used technologies and services:

* Java and Spring Framework on the backend.
* HTML, CSS , Material Design, Typescipt and Angular 2 on the front-end.
* PostgreSQL to store information about users, works, orders etc.
* Auth0 service for generating tokens for users.
* JWT technology for managing requests.
* Cloudinary cloud platform for keeping images from website.

.

2.2 User interface

This project will have user-friendly interface and look like book, text editor or online shop depending on what user is going to do.

2.3 User characteristics

* The writer: person who wants to publish his work or come up with a continuation of existing work;
* The reader: person who is interested in reading;
* The buyer: person who wants to buy masterpiece he likes

2.4 Assumptions и dependencies

Some problems with rendering in old versions of browsers may occur.

3. System requirements

3.1 Functional requirements

*3.1.1 Navigation bar on the top side of each web page*

* Fairy-Teller logo (by click user will be redirected to main page)
* Search bar (user may enter the key word and find results of the full-text search request )
* Profile button (user will visit the profile by clicking the button).
* Configuration button(user will be able to choose site language by clicking the button)

*3.1.2 Navigation bar on the left side of almost each web page*

* Buttons with genres of works. User will see the most popular works in chosen genre by clicking certain button.
* Toggle to include amateur’s works.

*3.1.3 Book appearance*

* Original cover of book
* Read button (will redirect user to the last bookmark user left).
* Overview button (will redirect user to chosen book page).

*3.1.4 Ordering section*

As the order of the book is individual, ordering section will provide:

* Dropdowns for font size and book format
* Carousel to choose cover user likes
* Order button (click will implement ordering the book).

3.2 Non-Functional requirements

* + 1. *Software Quality attributes*
* Security: will be and prevent server from unauthorized requests. Only authenticated users may change content of database.
* Performance: users don't want to wait for a long time watching the web page is loading. To avoid wasting precious time user must wait maximum 1 second to load any page on web app. Measures in time needed to load web page.
* Reliability: users should have a possibility to work with site 24/7. That’s why it’s important to provide such ability for them. Measures count of situations when user can't load web page.
* User-friendly interface: measures in user ability to intuitively understand the context and search the site without any problems.
  + 1. *General Patterns*
* Works of amateur artists are posted in full version
* Works of famous artists are posted in a short version.
* The continuation of work created by user which is not the author can be attached to this work.

*3.2.3 Role abilities*

* Unauthenticated users have a possibility to watch works of all artists.
* Authenticated users in addition have an opportunity to create and edit their own works; continue, comment and estimate other’s works; order books
* Admin may process orders, delete and block users.