Doğukan Arslan

SWE 573

Software Development Practice

02.01.2023

Pinner

GitHub: https://github.com/dogukanarslan/software-development-practice

AWS: http://18.191.213.23/

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2022 Fall semester project reported in this report, I declare that: - I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the 2022 Fall semester. - All the material that I am submitting related to my project (including but not limited to the project repository, the final project report, and supplementary documents) have been exclusively prepared by myself. - I have prepared this material individually without the assistance of anyone else with the exception of permitted peer assistance which I have explicitly disclosed in this report

Doğukan Arslan

Signature:

All work in this project is performed by me, Doğukan Arslan.

Table of Contents

Overview

Pinner is a web application which you can save and share resources that you gathered all around the web as a pin. In addition to sharing, you can also interact with other users by following them and liking/disliking pins that they shared. Development of Pinner is written using ExpressJS, web framework for NodeJS, and MongoDB, NoSQL database, for storing data. This project is deployed to Amazon Web Services EC2 machine as a Docker container.

Credentials of a general purpose user account

Mail: john@doe.com

Password: 123456

Software Requirements Specification

Functional

Authentication

1. If the current authentication token is not valid, the system shall redirect the user to the login page.
2. The system shall prevent registering with an existing email.
3. The system shall require user password at least 6 characters.

Create

1. When a user saves a pin, the system shall add it to pins page.
2. When a user creates a to-do, the system shall add it to the to-dos page.
3. When a user creates an account, the system shall validate if it is a valid email address.
4. The system shall store references of contents from different resources.
5. When a user saves content from any source, the system shall create a reference to the database.

List

1. The system shall sort records by descending order.
2. The system shall sort records by created at column.
3. The system shall list current user pins and all user pins separately.
4. The system shall allow the user to search for pins by title and description.

5. The system shall allow users to search for other users by name and surname.
6. The system shall allow the user to list all pins or pins that he/she follows.

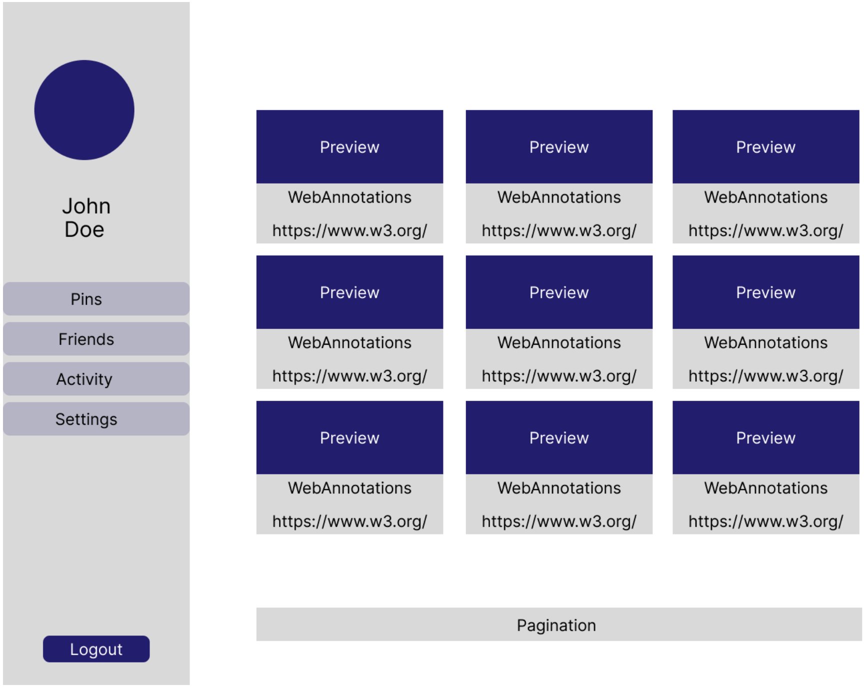
Interaction

1. When a user likes a pin, the system shall check if that post is previously disliked.
2. The system shall allow users to follow other users.
3. The system shall allow the user to add to-do on the existing pin.
4. The system shall have like and dislike functions for pins.
5. The system shall allow each post to be liked once by each user.
6. When a user deletes a to-do, the system shall redirect him to the to-dos page.

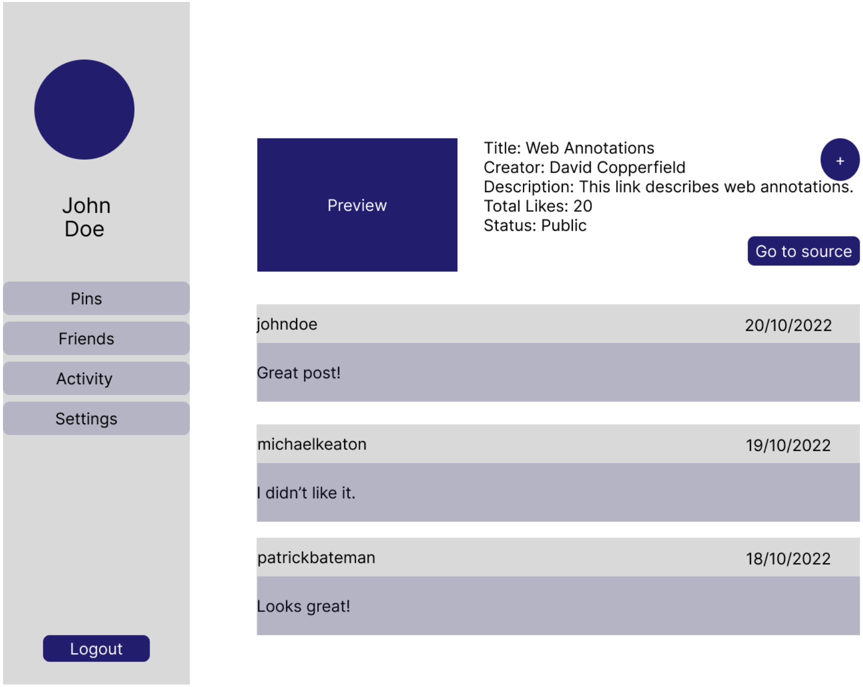
Non-Functional

1. The system shall have reusable components.
2. The request shall return a response in maximum 5 seconds.
3. The password of user accounts shall be hashed.
4. The authentication token shall be valid for 3 days

Design (Software and Mock-ups)



(First design of homepage)



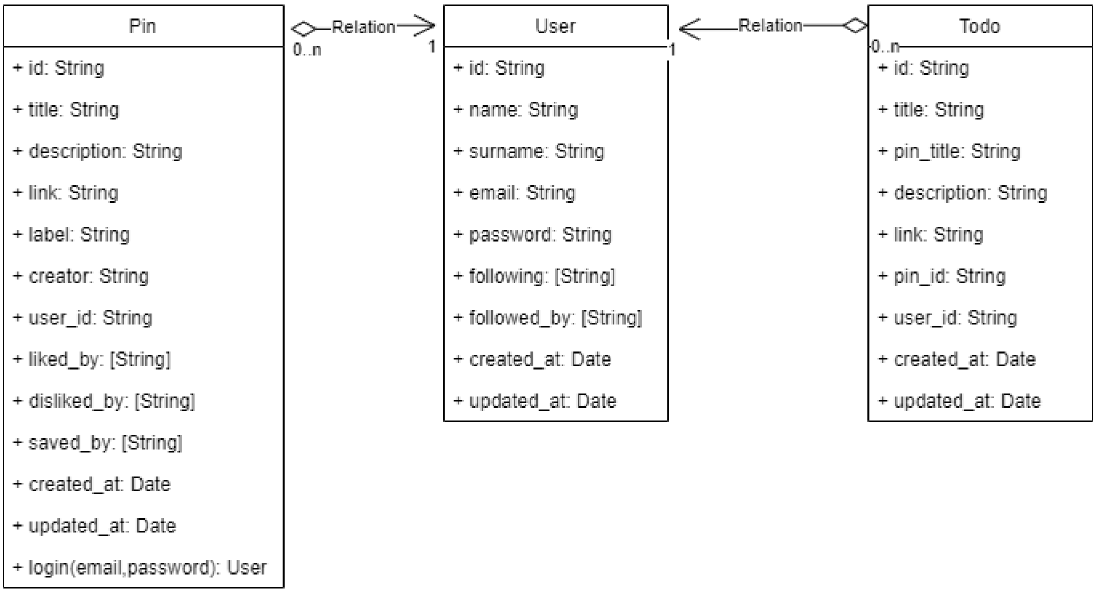
Title: Web Annotations
Creator: David Copperfield
Description: This link describes web annotations.
Total Likes: 20
Status: Public

johndoe 20/10/2022
Great post!

michaelkeaton 19/10/2022
I didn't like it.

patrickbateman 18/10/2022
Looks great!

(First design of pin detail page)



(UML diagram of models)

Status of Project

All requirements mentioned in software requirements specification part are completed, dockerized and deployed. There are features that planned in first mockups and didn’t implement during development with some considerations in mind. One of those features is deleting a pin. Since this is a collaboration project and users can save other users pins, deleting a pin may cause a problem for other users when they want to access it later. Another not implemented features is listing pins as a card, just like in mockups. This would take up too much space and it would take much more time to scroll down when there are too many pages. Instead of cards, listing data in a table was more suitable. And the data listed on table is not enough for a user, he/she can always visit details page for more information. Finally, commenting on pins that are available on first mockups, is not implemented, however current architecture of this project is quite suitable for adding it.

Status of Deployment

This project is dockerized and then deployed to AWS, EC2 machine.

Deployed URL: http://18.191.213.23/

System Manual

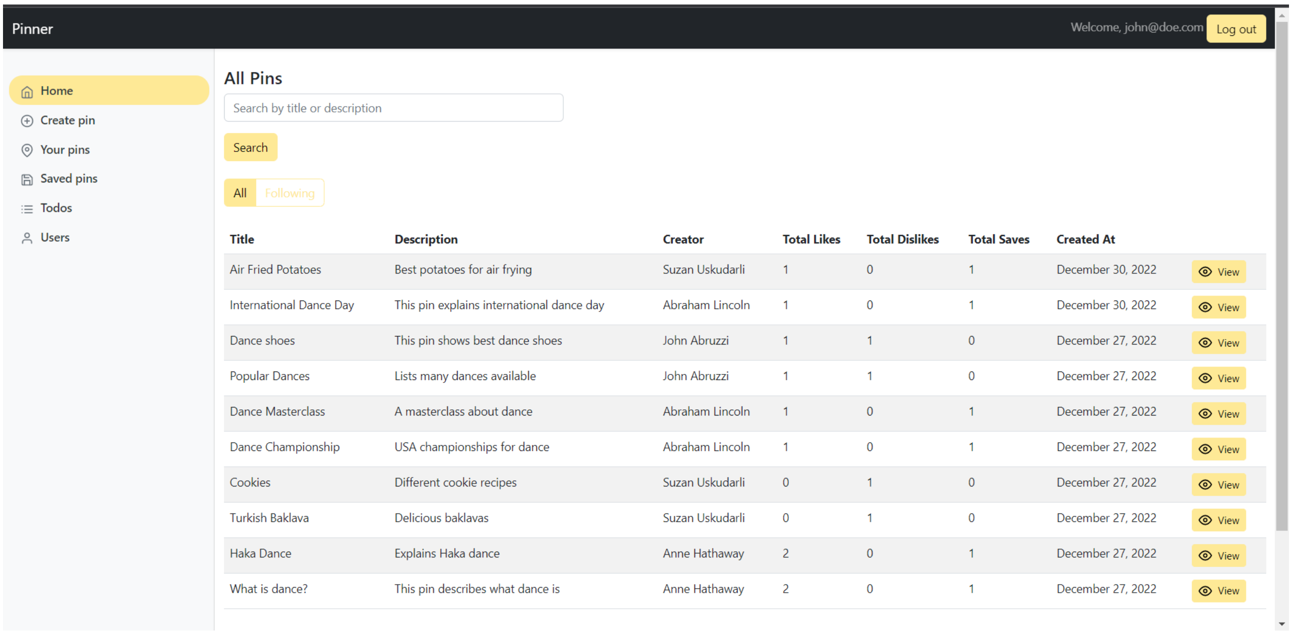
Pinner is a NodeJS project, to run this project locally you need to have NodeJS installed on your local system. You can use this link to install NodeJS. Also, MongoDB must be installed on your system locally. You can install it with this link Alternatively, you can use MongoDB cloud database.

Before running the project locally, database URI and secret key must be added to .env file. To make this process easier, you can duplicate .env.example file and rename it as .env. After that, use your database URI, that you obtained from locally installed MongoDB, as MONGODB_URI variable. You should use another database for test cases. You can do it by initializing MONDB_TEST_URI variable with your existing database URI and adding /test-database to end of it. After completing these prerequisites, you should be able to run the project locally.

Since this project is dockerized, if you want to run project as a Docker container without installing any local dependencies such as NodeJS or MongoDB. To use docker, it should be installed on your local machine. You can install it with this link. Before running project as a docker container, Docker client must bu running. You can run docker-compose up –d command in the project directory and Docker will create a container for the project and another container for the database and link them together. After it is done, the project should be available at port 80.

User Manual

Pinner is a web application where you can save resources as a pin. When you open the web application, you create an account with an email. This email should be unique in the system, you can’t create more than one account with the same email. After creating an account, you are logged in the system ama at homepage you can see pins that are created by other users. In homepage, you have two options for listing pins, one is listing pins from users that you follow, second is listing pins from all users. Since listing all pins would be chaos, pins from follower’s feature is implemented. If you are looking for a specific pin, you can search it by title and description. Pins are listed as a table, and you can view details of a pin by clicking the view button at the last column of the table.



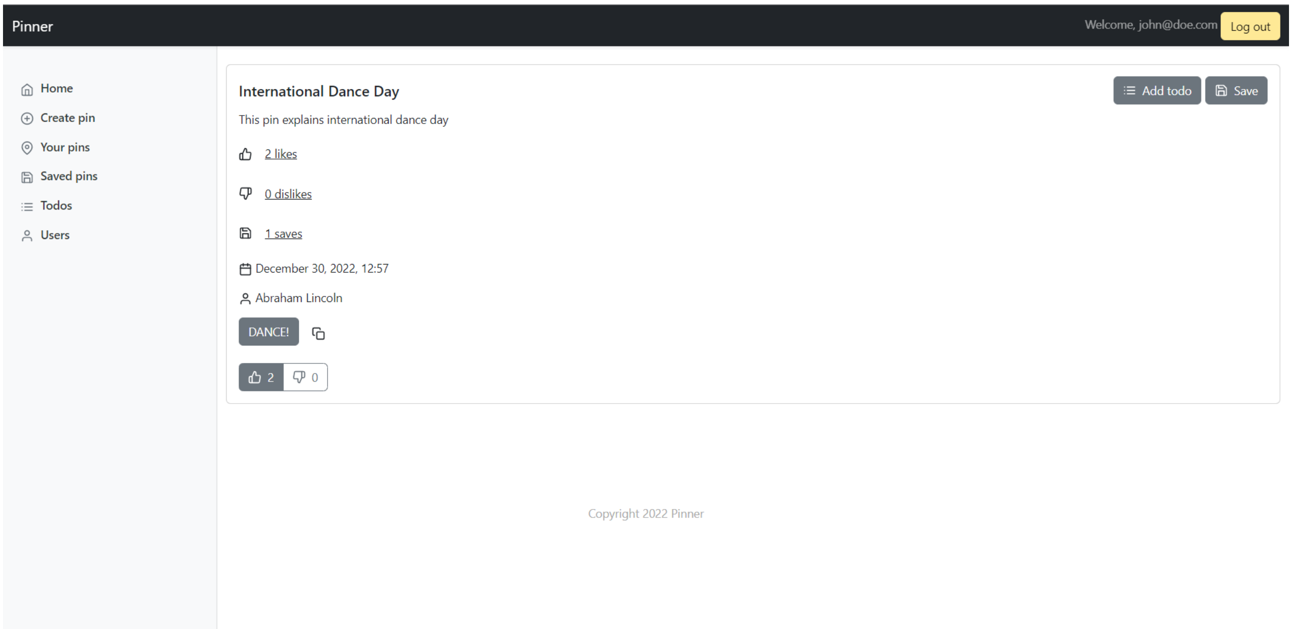
All Pins

Search by title or description

Title	Description	Creator	Total Likes	Total Dislikes	Total Saves	Created At	
Air Fried Potatoes	Best potatoes for air frying	Suzan Uskudarli	1	0	1	December 30, 2022	View
International Dance Day	This pin explains international dance day	Abraham Lincoln	1	0	1	December 30, 2022	View
Dance shoes	This pin shows best dance shoes	John Abruzzi	1	1	0	December 27, 2022	View
Popular Dances	Lists many dances available	John Abruzzi	1	1	0	December 27, 2022	View
Dance Masterclass	A masterclass about dance	Abraham Lincoln	1	0	1	December 27, 2022	View
Dance Championship	USA championships for dance	Abraham Lincoln	1	0	1	December 27, 2022	View
Cookies	Different cookie recipes	Suzan Uskudarli	0	1	0	December 27, 2022	View
Turkish Baklava	Delicious baklavas	Suzan Uskudarli	0	1	0	December 27, 2022	View
Haka Dance	Explains Haka dance	Anne Hathaway	2	0	1	December 27, 2022	View
What is dance?	This pin describes what dance is	Anne Hathaway	2	0	1	December 27, 2022	View

(Screenshot of homepage)

In detail, you can edit the pin if you own it, otherwise you can't. You can save that pin for accessing it later and you can create a to-do with that pin. Saved pins and to-dos are listed on different pages, you can access them from the sidebar. Also, you can interact with a pin by liking or disliking it. Users that liked or disliked are listed in a modal that opens when you click on the like and dislike counts.



International Dance Day

This pin explains international dance day

2 likes

0 dislikes

1 saves

December 30, 2022, 12:57

Abraham Lincoln

DANCE!

2 0

Copyright 2022 Pinner

(Screenshot of pin detail page)

At users page, you can search for other users that are on the Pinner by name and surname and you can see details of it if you click on view button. In detail, you can follow or unfollow a user. This following is used when you are on homepage and listing all pins or pins from users that you follow. Also, you can see followers of that user and users that he/she follows when you click on following on followers count.

Tests

Test cases are implemented using Jest and there are tests for creating an account, logging in with that account and creating a new pin. Each test checks if the response has a correct status code and correct response body. To run a test locally, you can run npm run dev command on project directory and it will run all tests one by one. When you run a test, it uses a database that is separated from the original database to prevent polluting the production data. You can define the database in .env file with MONGODB_TEST_URI.

Introduction Video

This video is a product walk-through to have a better understanding of usability of project.

Introduction Video Google Drive Link

Repository

This project is developed using git as a version control system and this git repository is persisted on remote using GitHub
GitHub Link

Technologies

This project is built mainly using ExpressJS. As a template engine EJS is used to make it easier to develop html pages. On database side, MongoDB is used as a NoSQL database and database relations are managed using Mongoose. For the unit and user tests, Jest test framework is used. Authentication process is completed using JSON Web Tokens (JWT). Finally, project containers are built with Docker and they are deployed on AWS EC2 machine.

- ExpressJS
- EJS
- MongoDB
- Jest
- Docker
- JWT
- AWS

References

1. This video from Youtube by NetNinja channel was quite helpful for authentication structure.
2. ExpressJS documentation is highly used.
3. Mongoose documentation is highly used.
4. MongoDB documentation is highly used.
5. Jest documentation is highly used.