Project Proposal based on Research Paradigm, initial literature review schedule and risk analysis.

Claudinea de Almeida

Academic Paper Management on the Cloud

Abstract

Nowadays, with a technological expansion of cloud computing, there has been a giant expansion of the possibilities of accessing and sharing data in general and a demystification of knowledge.

The best way to share and disseminate knowledge is to create a management system for this data. On the academic side, when you're looking for an idea, or simply getting to know technological innovations, it can become a nightmare. Despite finding a lot on the internet, it is still very restricted to academic proposals and master's and doctoral works.

The idea is based on the development of a web portal, where authors of master's theses and course conclusion works can be published in a community database, where everyone can have free access to research works.

This project will allow users to make their theses public, organizing them by category, even with the different ways of exposing their article.

Currently, systems of this nature are lacking, with few exceptions, and even the existing ones do not have the flexibility to organize the publication by topic, and even free of charge, all based on the most current technologies on the market, which will be discussed later in this document.

Keywords:

Thesis management system, Cloud Computing, API, Publishing Articles

Contents

Introduction

The main idea of this project is to create a system, based on cloud computing, focused on web page and mobile application, which allows students and professors to publish their articles, theses, final projects of course conclusion automatically and independently.

The system will facilitate the opening of articles to the general public with unrestricted access, organizing publications by categories and making university articles available to everyone. This makes it easier for master's students to have an idea of published articles that will serve as a reference for their future projects.

The System will be developed with the most current technologies and with a simple and understandable interface so that all students, teachers, and readers in general can navigate without complications inherent to information technology.

As a software engineer, my mission is to specify details of how the system works and the best approach so that the result meets users' expectations.

Research question and Rational

1. How can I allow unrestricted system access and uploading without affecting software performance?
2. How to categorize the different types of thesis topics and final projects?

The first question can be answered by utilizing the best qualities found in using cloud computing, which allows scalability to handle the changing needs of an application within the confines of the infrastructure by statically adding or removing resources to meet demands. applications, if necessary.

For the second question, the answer will be the use of tags, which allow the merging of related articles.

Literature Review

In this session, basic concepts of cloud computing and its particularities that allowed the adequate performance of the proposed system will be discussed. The subject of tags will still be discussed and how we will approach this algorithm to properly categorize published articles.

When it comes to cloud computing, what can be kept in mind, speaking at a high level, is that it makes the issue of physical structure and the location of the servers where the system will be stored flexible, one of the characteristics is that it has elasticity and scalability that allow the server's capacity to be expanded during peak usage times, allowing the system to function normally, without affecting its performance and being invisible to the eyes of the end user.

Regarding the database, NoSQL will be used, which speeds up the inclusion of data and improves the speed in the search for data, without the need for a complex relationship structure used in SQL databases.

In addition, the search will be done with API technology, which simplifies and speeds up the search for data, thus allowing an extra help in terms of performance.

Tags are a way of grouping related subjects, in this way, we will be able to organize the published articles in a more specific way, both from the point of view of who is doing the publication and from the side of the future reader, looking for more consistent related subjects.

Technologies

This program will be created using Node.js environment, JavaScript as back-end language, MongoDB Atlas as cloud database, Express as server, HTML and Ejs as scripting language and front-end, Bootstrap as framework, GitHub as repository and version control, Postman for API testing.

Node.js

Node.js can be defined as a server-side JavaScript execution environment.

This means that with Node.js it is possible to create JavaScript applications to run as a standalone application on a machine, not depending on a browser to run, as we are used to.

MongoDB Atlas

It is a global cloud document database service for modern applications. Basically, you only care about managing the data that will be there, all the infrastructure and maintenance of the machines, as well as the security of all this is up to them.

NoSQL

NoSql databases use a variety of data models to access and manage data. These database types are specifically optimized for applications that require large data models, low latency, and flexibility. These requirements are met by relaxing some of the other databases' data consistency restrictions.

Express

Having its initial version released in 2010, Express.js (or just Express) is a framework for developing JavaScript applications with Node.js.

HTML and Ejs

HTML (Hypertext Markup Language) is used to design the structure of a web page and its content. HTML is not technically programming languages like C++, JAVA, or python. It is a markup language and ultimately provides declarative instructions to the browser.

EJS simply means inline JavaScript. It is a simple template engine or language. EJS has its own syntax and defined tags. It offers an easier way to generate HTML dynamically.

Bootstrap

Bootstrap is a front-end framework that provides CSS frameworks to create responsive websites and apps quickly and simply. Also, it can handle both desktop websites and mobile pages.

GitHub

GitHub is a cloud-based service that hosts a version control system (VCS) called Git. It allows developers to collaborate and make changes to shared projects while keeping a detailed record of their progress.

Postman

Postman is a tool that supports the documentation of requests made by the API. It has an environment for documentation, execution of API tests and requests in general.

API

APIs are a set of patterns that are part of an interface and that allow the creation of platforms in a simpler and more practical way for developers. From the APIs it is possible to create software, applications, programs and different platforms.

Architecture

As far we will using MongoDB as a Database, we will have the architecture like described below:

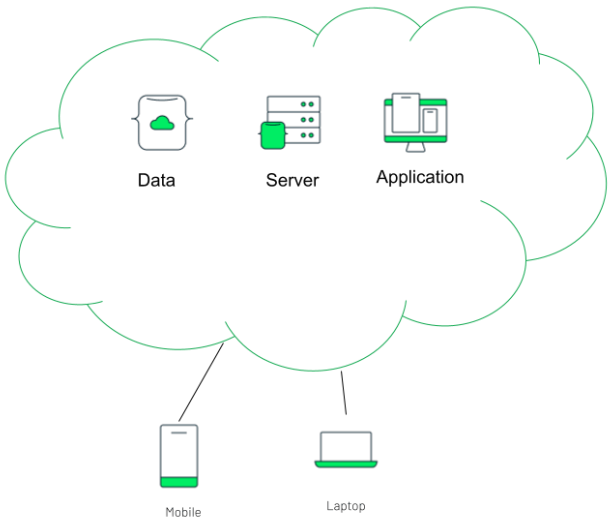

Figure 1- SaaS and MongoDB (MongoDB. n.d.)

Related Work

A similar project was found but that focuses on the university's control over the upload of articles and approval of the topic desired by the professor, this project in particular focuses on the search for example articles, not on the approval of article proposals by the professor. (Khalil, Y., 2018)

There is the website http://www.academia.edu, which has something similar, but they don't organize by tags, so the subjects are often mixed and confusing, besides they charge for the full article published, you don't have free access total.

Proposed Work

The proposal consists of creating a system for registering student articles in two main points:

1. Uploading articles
2. Creating tags for grouping related topics

Use case Uploading articles

Item	**Description**
Use Case ID	UCAD01
Use Case Name	Uploading articles
Actor:	Student/Professor
Trigger	Before Start the upload of the article
Business Role	Title cannot be empty
Description	The user case uploads the article
Normal Flow	The user inserts the data, upload the pdf article and save
Alternate Flow	N/A
Cross reference	N/A

Use case Creating tags for grouping related topics

Item	**Description**
Use Case ID	UCAD02

Use Case Name	Creating tags
Actor:	Student/Professor
Trigger	While uploading the article
Business Role	Tag info cannot be empty
Description	The user chooses the tag or type new one before uploads the article
Normal Flow	The user selects or type the tag and save
Alternate Flow	N/A
Cross reference	UCAD01

Proposed Implementation

An online system was created based on a simple and friendly screen that allows searching, viewing the data already registered, adding a button, updating, and deleting data.

Home page

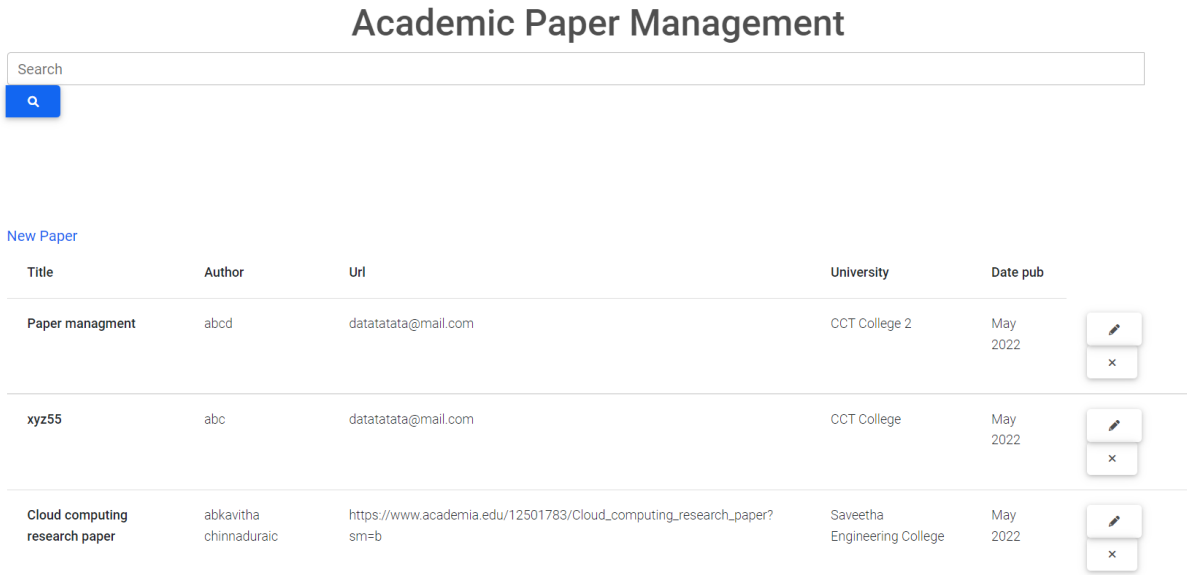

Figure 2 - Home Page

Project Explorer

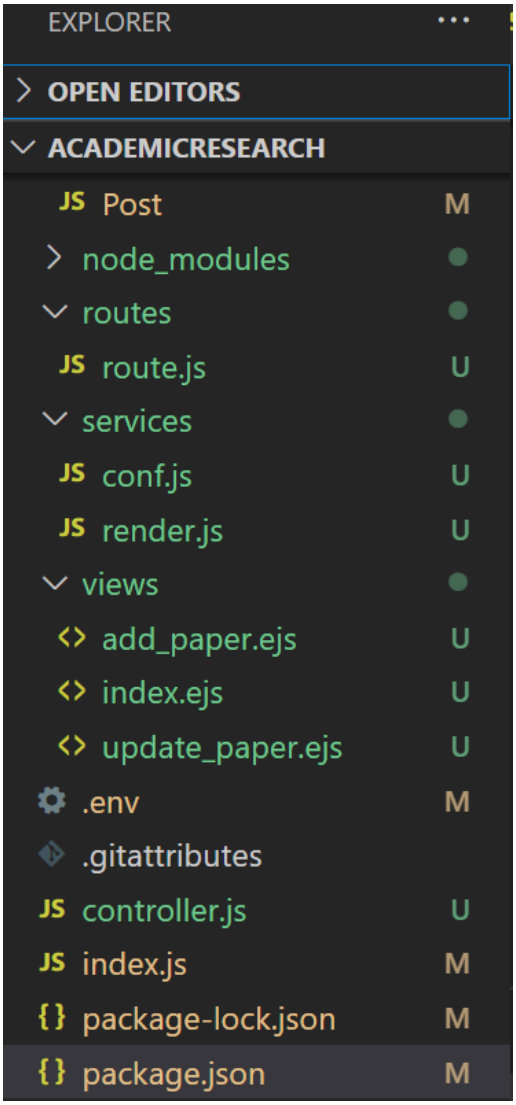

Figure 3 – Project Explorer

Package.json

In this file are listed the dependencies and libraries used in the system.

```
{
  "name": "academicresearch",
  "version": "1.0.0",
  "description": "Node.js, Express, MongoDB Altas web app",
  "main": "index.js",
  "scripts": {
    "devStart": "nodemon index.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
```

```
  "repository": {
    "type": "git",
    "url": "academicresearch"
  },
  "author": "Claudinea de Almeida",
  "license": "ISC",
  "dependencies": {
    "axios": "^0.26.1",
    "body-parser": "^1.20.0",
    "dotenv": "^16.0.0",
    "ejs": "^3.1.6",
    "express": "^4.17.3",
    "express-session": "^1.17.2",
    "github": "^14.0.0",
    "mdb-ui-kit": "^3.11.0",
    "mongoose": "^6.2.10",
    "morgan": "^1.10.0",
    "path": "^0.12.7"
  },
  "devDependencies": {
    "nodemon": "^2.0.15"
  }
}
```


Proposed Evaluation

Research Timeline

The Gantt chart shows the basic timeline of the project

Task	Feb	Mar	April	May
Proposal initiation				
Learning concepts				
Dataset preparation				
Study of Program Language, tools, and libraries				
Developing Uploading Article				
Developing Tag				
Developing search				
Evaluation of results				

Figure 4 – Project Gantt chart

Conclusion

References

Docs.mongodb.com. 2021. [online] Available at: <https://docs.mongodb.com/manual/text-search/> [Accessed 27 December 2021].

W3schools.com. 2021. *Node.js MongoDB Find*. [online] Available at: <https://www.w3schools.com/nodejs/nodejs_mongodb_find.asp> [Accessed 27 December 2021].

Youtu.be. 2021. *Build A REST API With Node.js, Express, & MongoDB - Quick*. [online] Available at: <https://youtu.be/fgTGADljAeg> [Accessed 27 December 2021].

Youtu.be. 2021. *Build A Restful Api With Node.js Express & MongoDB | Rest Api Tutorial*. [online] Available at: <https://youtu.be/vjf774RKrLc> [Accessed 27 December 2021].

Youtu.be. 2021. *Como criar um mecanismo de busca em Node.js e MongoDB*. [online] Available at: <https://youtu.be/qLO8Q870fmc> [Accessed 27 December 2021].

Khalil, Y., 2018. *Development of a web-based system for thesis and project proposal management*. [online] Academia.edu. Available at: <https://www.academia.edu/70051130/Development_of_a_web_based_system_for_thesis_and_project_proposal_management> [Accessed 10 April 2022].

Agyepong, S., 2011. *MSEARCH: A Mobile Search Service*. [online] Academia.edu. Available at: <https://www.academia.edu/70362064/MSEARCH_A_Mobile_Search_Service> [Accessed 5 March 2022].

Dasic, P. and Crvenkovic, B., 2016. *Applications of the Search as a Service (SaaS)*. [online] Academia.edu. Available at: <https://www.academia.edu/29837306/Applications_of_the_Search_as_a_Service_SaaS_> [Accessed 5 March 2022].

Dašić, P., Dašić, J. and Crvenković, B., 2016. *Service Models for Cloud Computing: Search as a Service (SaaS)*. [online] Academia.edu. Available at: <https://www.academia.edu/29576005/Service_Models_for_Cloud_Computing_Search_as_a_Service_SaaS_?pop_sutd=false> [Accessed 5 March 2022].

Kanade, S. and Manza, R., 2019. *A Comprehensive Study on Multi Tenancy in SAAS Applications*. [online] Academia.edu. Available at: <https://www.academia.edu/61390511/A_Comprehensive_Study_on_Multi_Tenancy_in_SAAS_Applications> [Accessed 5 March 2022].

s, s., 2012. *CLOUD COMPUTING : SAAS*. [online] Academia.edu. Available at: <https://www.academia.edu/21491005/CLOUD_COMPUTING_SAAS> [Accessed 5 March 2022].

Samir, A., n.d. *An Aspect-Oriented Approach for SaaS Application Customization*. [online] Academia.edu. Available at: <https://www.academia.edu/65168311/An_Aspect_Oriented_Approach_for_SaaS_Application_Customization> [Accessed 5 March 2022].

MongoDB. n.d. *What Is SaaS? Software-as-a-Service Explained | MongoDB*. [online] Available at: <https://www.mongodb.com/cloud-explained/saas-software-as-a-service> [Accessed 1 April 2022].