

pragoonsoni /
ResearchHaven-SIH-FRONTEND

<> Code

Pull requests

Actions

Projects

Security

Insights



Problem Statement By UGC - Development of system to avoid duplicity of Research Projects submitted to various funding agencies

[webcrawlers.tech](#)

MIT license

4 stars 1 fork 1 watching 1 Branch 0 Tags Activity

Public repository

1 Branch 0 Tags Go to file Go to file Add file Code

pragoonsoni Merge pull request #1 from pragoonsoni/main e0bd64c · 2 years ago

public	improved theme	2 years ago
src	final commit	2 years ago
.gitignore	added homePage	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Update README.md	2 years ago
package-lock.json	Added download pdf option	2 years ago
package.json	Added download pdf option	2 years ago

README MIT license

ResearchHaven



Inspiration

Our problem statement is related to the **duplicity of research proposals** submitted to various funding agencies.

- As we all know, a typical research proposal can take **weeks to months to even years** to get approved, and one may **not even get to know** that their proposal is rejected and hold onto an empty hope. A very common cause for this is the fact that a proposal has to go through **numerous stages** before getting approved.
- First stage being **submitting the proposal** in a selected format and deciding all the relevant details like **selecting a specific problem** to solve or **deciding which funding organization** to specifically apply to.
- Next, the proposal is **checked for plagiarism** and then finally, it is **peer reviewed** and pushed for funding approval. As we can see, each stage of the process is important, however the first stage of formatting is not as pertinent.
- However, **paraphrasing and a lot of other malpractices** can go unnoticed when it comes to just plagiarism checks on a basic level. This leads to proposals which **get stuck in this stage** and not get approved.
- The **main drawback** is that the researcher doesn't get any report as to what caused their proposal to not be approved. Consequently, this results in **higher time and resource consumption with no fruitful results**.
- Due to this, at times, a **deserving candidate** might **not get funding**, thus slowing down the growth of the research field.
- We hope to solve such problems with our project.



About the Product

- A **fully functional Web App** that acts as a central platform to apply for funding for your projects and **get them approved or rejected**.
- We aim to help **create and foster an environment of responsible and competitive research** in our country.
- We hope to connect the **innovative minds and their ideas** within our country to **the people who can give them the resources to actualize these ideas**. For the same, we have tried to keep our UI/UX as simple as possible and are always open to any constructive feedback.



What it does ?

- Our **base idea** is to provide users with a fully functional Web App that acts a **central platform for applying for funding** that aims to use the current developments in **decentralized storage architecture** to provide them with a platform that allows them to upload or create their draft proposals and submit them for plagiarism checks, clearing which they will be submitted for consideration to fund.
- We use the **modern MERN stack** to store our user's details and drafts after encrypting the data via **bcryptjs**. We require a one-time registration on our software, and then they can access the platform at their leisure.
- Once the users have submitted a draft proposal, we use a **custom-built, state-of-the-art NLP algorithm** to check for plagiarism on **3 levels**, and if the proposal clears all of them, it is forwarded th the relevant

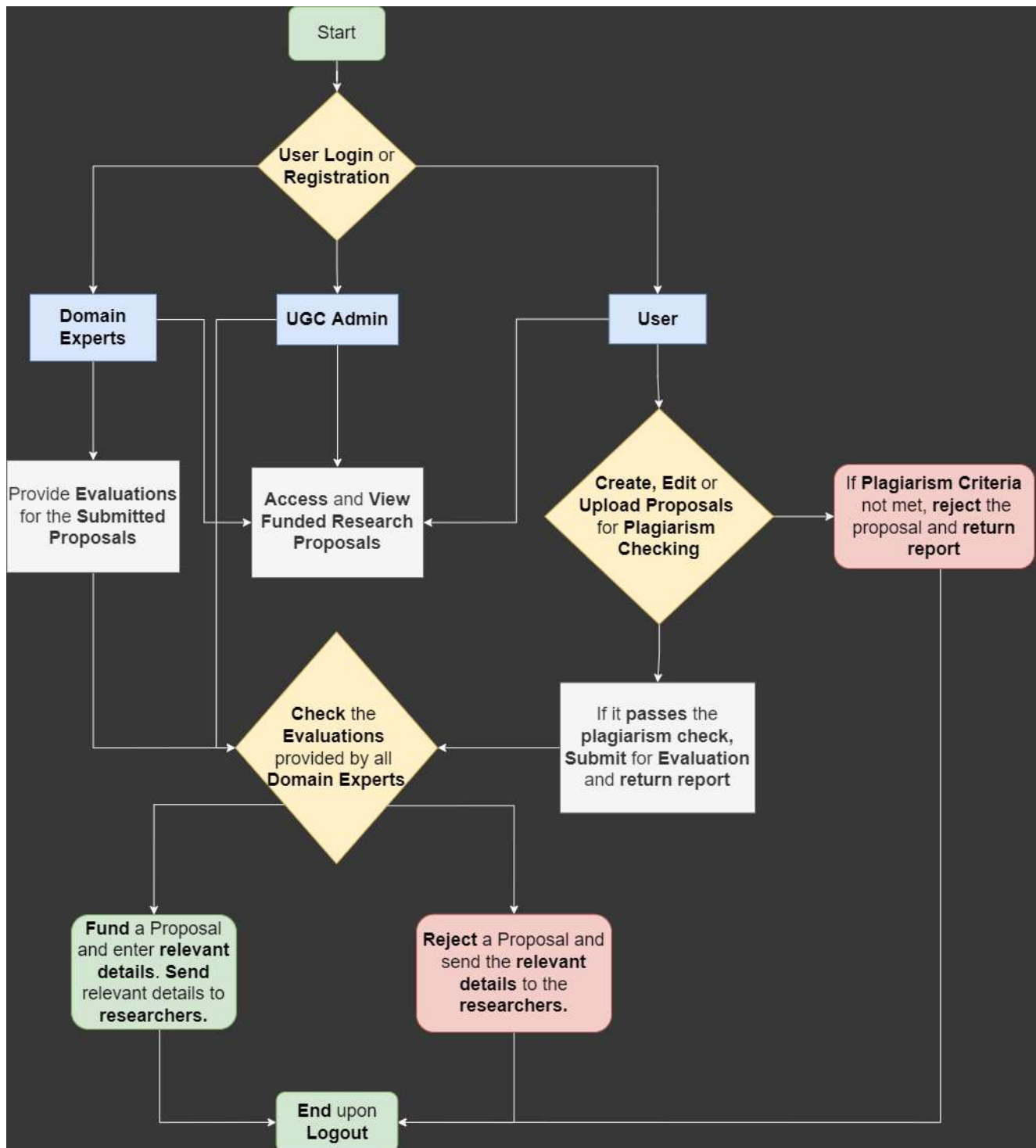
authorities for consideration for funding.

How we built it ?

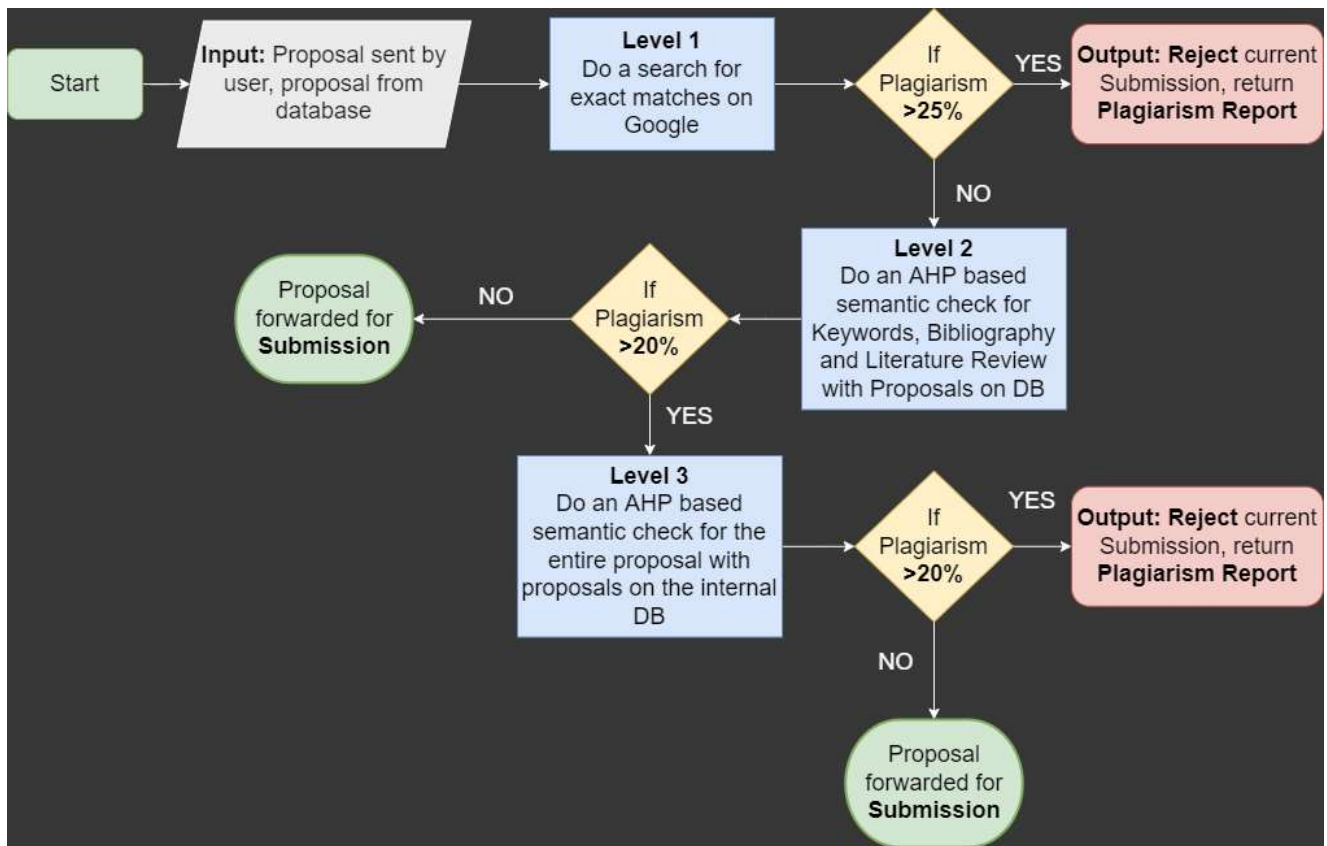
- **Front-End:** We used ReactJS and Chakra UI to create a simple, elegant and minimalist user interface that meets our design requirements and maximises efficiency.
- **Back-End:** We used several JS 'npm' packages, as well as NodeJS as a runtime environment to create an effective and effortless back-end module.
- **Storage:** We used the MongoDB database to store all the data except the funded papers, which are further pushed on the Ethereum Blockchain.
- **NLP Model:** Most importantly, we used various python libraries like NLTK and YAKE along with implementing the AHP algorithm to give **unique weights** to every section of the Research Proposal.

How it works ?

Overall Work Flow



NLP Work Flow and Levels Explanation



🧐 Challenges we ran into

- Ensuring that at no place did we access our **user's credentials**.
- Coming up with a sound **business plan** for a project that can very well be in the public service domain.
- Overcoming various errors as we learned how to use the **AHP Algorithm** for the first time.
- Overcoming various **CORS** errors by trial and error when trying to integrate the **back-end API** with the front-end architecture.
- Reorienting ourselves constantly as we received feedback for the different new features we need to implement at each review.
- We faced cors error and we tried to resolve it.

🌟 Accomplishments that we're proud of

- Our **USP** that we made **innovative use** of **AHP** and **NLP libraries** to create a custom, extremely efficient Plagiarism checker.
- Combine that with the **minimalistic design** and **efficient software** development, and we have the perfect funding applications submission software for our country.
- We have created a system in which the drafts are **constantly rendered** and the they are never stored without encryption.
- Moreover, ensuring smooth working of such a large project with these many components is a testament to our team spirit and target orientation.



What we learned

- Learned how to use the **AHP Algorithm**.
- Various new **ReactJS** and **NPM** packages that we used at various places in the front and back-end.
- Implementing **NodeJS** and **ExpressJS** to store and retrieve data in such a complex system where we have made **over 50 unique APIs**.



Features

Trustless

Releases

No releases published

Packages

No packages published

Contributors 3



nandurijv Nanduri Jayant Vishnu



alok27a Alok Mathur



pragoonsoni Pragoon Soni

Languages

JavaScript 97.2% CSS 2.1% HTML 0.7%