

Napolean Suite

MINOR PROJECT SYNOPSIS

BACHELOR OF TECHNOLOGY

Information Technology

SUBMITTED BY:

Priyanshu Tiwari(1921127/1905380)

Tanuj Sharma (1921106/1905406)

Shubham Savarn (1921138/1905400)

SUBMITTED TO:

Dr. KS Mann

JAN-JULY 2022



GURU NANAK DEV ENGINEERING COLLEGE

LUDHIANA-141006 INDIA

Contents

1	Introduction	3
2	Objectives	4
3	Methodology	5
4	Work Flow	6
5	Facilities required	7
6	Expected Outcome	8
7	References	9

1 Introduction

Most of the major IT corporations are leveraging online coding competition to judge the pressure handling and fundamentals of upcoming software engineers. This has led to a significant increase in the number of online judges and coding competition. Most of the students are now confused, which platform they should opt and how to approach these coding contests on time, every time. This is where Napoleon Suite comes into the picture.

Napoleon Suite suite is a collection of extensions, APIs, bots and web app aimed to simplify competitive programming. With Napoleon students can observe, compare, shortlist and outperform in these online judges and compare the improvements and achievements with their peers in a healthy environment.

Technologies that we'll be using in this project will be Python, Javascript, Node, Flask, Selenium, VueJS, Tailwind.

The frontend of a software program or website is everything with which the user interacts. From a user standpoint, the frontend is synonymous with the user interface. Technologies employed for frontend is Vue along with Tailwind and Chart Js

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Flask is a lightweight Web Server Gateway Interface WSGI web application framework that was created to make getting started easy and making it easy for new beginners.

Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation. It provides a playback tool for authoring functional tests without the need to learn a test scripting language.

Vue.js is an open-source model-view-viewmodel front end JavaScript framework for building user interfaces and single-page applications.

Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all of the building blocks you need to build bespoke designs without any annoying opinionated styles you have to fight to override.

2 Objectives

The main objective is to create a platform on which students can easily select and prepare for online coding competition in the best possible way.

The key objectives of Napoleon Suites are:

1. **Looking all the competitive profiles at a glance.**
2. **Get updates about latest programming contests.**
3. **Getting all the updates through email newsletter and push notification.**
4. **Fetching global and local leadership .**
5. **VS code extension to speed up local development.**
6. **Chrome extension to view upcoming contest on-the-go.**
7. **Standalone REST API.**

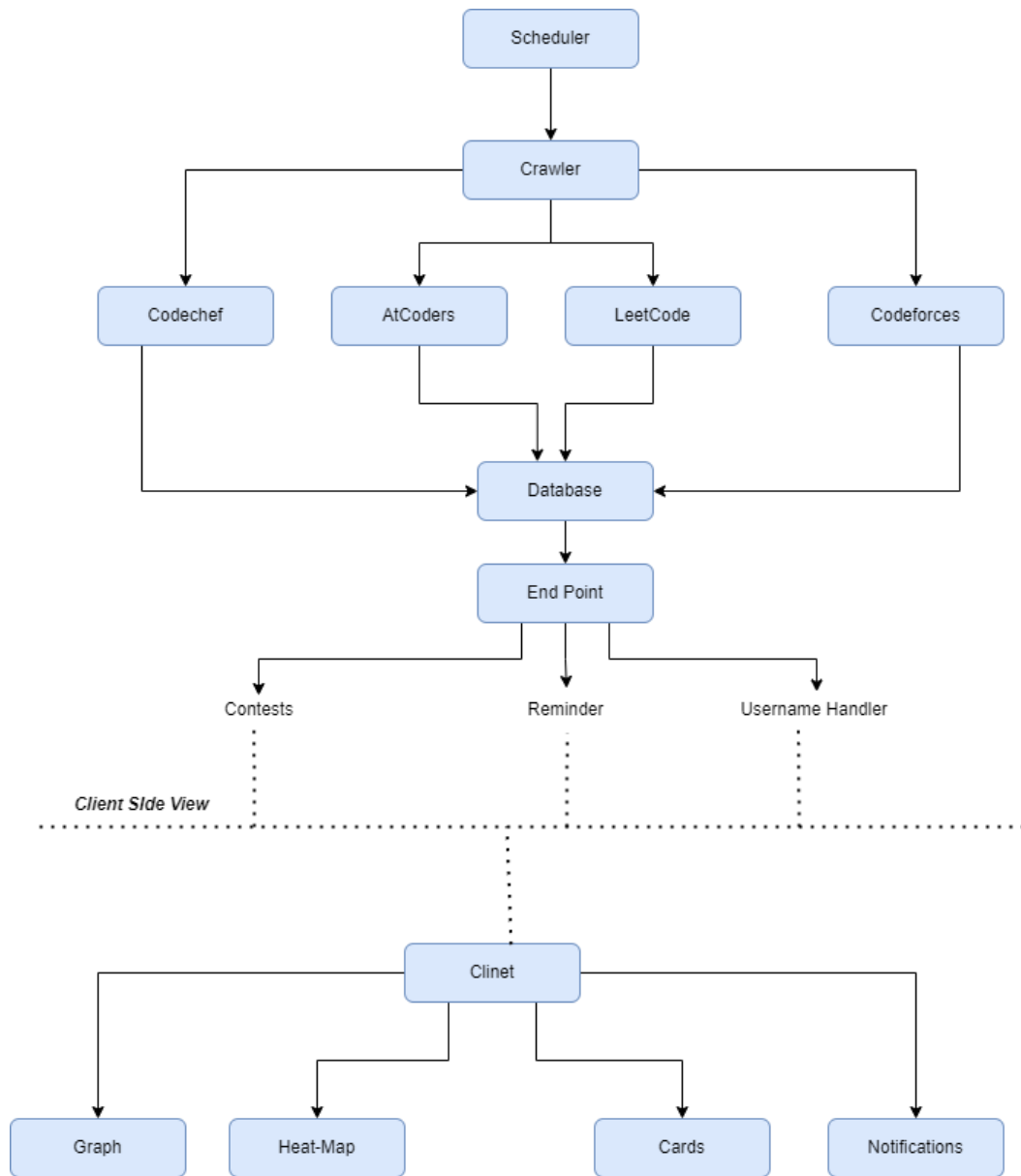
3 Methodology

In the first step we will scrape the data from various resources using a crawler built in Python with Selenium. We will store this data in our database and create pipeline with a cronjob in every six hours.

Now we will deliver all the extracted data through our SPA using VueJS. We will use Workbox 6.0 to convert our SPA into a Progressive Web Application and natively support push notification.

- **Web Scrapping** Web scraping is an automatic method to obtain large amounts of data from websites.
- **Cronjobs in recurrent pipelines** A cron job is normally used to schedule a job that is executed periodically. In our case we use a cronjob to run our python script that will fetch, extract unstructured HTML data, validate it and save it in our required database.
- **User interfaces** Building user friendly interfaces that brings meaning to our extracted data and visualizing it through various tables, charts and graphs.

4 Work Flow



5 Facilities required

- **Software Requirements**

- Vue, Tailwind, ChartJS, Babel, GSAP, Node
- Flask, Postgre, Selenium, Python
- Git, GitHub, CodeQL, VS Code
- NGINX, PM2, Travis, Certbot

- **Hardware Requirements**

- Any modern computer system supporting modern web browser.

6 Expected Outcome

- Responsive, minimalistic user interface with clutter-free user experience.
- Powerfull REST API that can power other third-party applications.
- Healthy competitive environment with 'freindly competition' among peers and making competitive programming a constructive habit.

7 References

- [1] <https://www.analyticsvidhya.com/blog/2020/08/web-scraping-selenium-with-python/>
- [2] <https://pypi.org/project/bs4/>
- [3] <https://nuxtjs.org/>
- [4] <https://flask.palletsprojects.com/en/2.1.x/tutorial/>
- [5] <https://jasonwatmore.com/post/2019/11/18/react-nodejs-on-aws-how-to-deploy-a-mern-stack-app-to-amazon-ec2>
- [6] <https://docs.github.com/en/actions>
- [7] <https://www.postgresql.org/docs/12/functions-json.html>
- [9] <https://www.geeksforgeeks.org/how-to-use-the-javascript-fetch-api-to-get-data/>