LET'S COMPETE

17.04.2022

GROUP 9

CSN 254: SOFTWARE ENGINEERING

GitHub Repository:

https://github.com/priyansh472/-Let-sCompete-Learning-and-Competition-Platform-

Project Proposal

We intend to create a website where people across the world can take part in events like hackathons, quizzes and multiple cultural events that are happening around them as well as grab opportunities including scholarship, internships; moreover, various job opportunities, workshops, seminars, and copious other opportunities which might be left off and some of them could prove to be particularly important for their personal and professional development.

Technology Used

· **Database** -	SQLite
· **Web-Based Interface** -	Js, HTML & CSS
· **Backend** -	Django

Key Features

Let's Compete provides users a single platform to take part in various events such as hackathons, quizzes and cultural events that are happening around them and get opportunities including scholarships, internships; moreover, different job opportunities, workshops, seminars which can be important for their professional and personal development.

Some of the basic features of our website will be:

- Filtered list of current and upcoming events will be shown to the user.
- Specifications (like date, eligibility etc.) and criteria related to every event will be provided in detail.
- People will be able to register for upcoming events that will provide great exposure to the competitive nature of different fields.

- Organizations can register themselves to provide a window of opportunity to interested people.

The website requires a user to make an account on the website through registration, after which they can customize their profile. The user can then access the features of Let's Compete and explore different competitions.

Design Diagram

Flow Chart
Use Case Diagram
Class Diagram

Description of Implementation

Development Process

Our team will be using a modified iterative waterfall model to develop this software. We chose this development method because this software has very specific and well-defined needs by the users, and these are not going to change. The team may concentrate on one area at a time, especially since some members in the team have less experience in coding and in large scale software development. A modified waterfall model allows us to spend more time on requirements, understanding the design and on developing a better system.

1. Outline Plan

Milestone 1 (05th March - 9th March) -

Learning to integrate different tools for our project

Milestone 2 (9th March - 14th March) -

Feasibility Study Report

Milestone 3 (14th March -20thMarch) -

Requirements Analysis Report

Milestone 4 (20th March - 26th March) -

Software Design Phase

Milestone 5 (26th March - 12th April) -

Development/coding Phase.

First 7 days half of the team (front-end team) will work on the user interface of the website while the back-end team will work on creating the server. The other 7 days back-end team will work on connecting databases and APIs to the website and UI team will work on the rest of interface designing.

Milestone 6 (14th April - 16th April) -

Deployment and Testing of the project.

2. Coding and Implementation Phase

Frontend

1. We have implemented a Banner Slider.
2. We have a navigation bar which is always present at the top of our website for easy access.
3. Owl carousel is used as running images.
4. We have implemented Search functionality so that a person in need of some particular product can have a faster access to the product.
5. We can register ourselves as a user or as an organization in order to register ourselves for the event or in case of organization we can upload our events for others so that the eligible candidates can register.
6. We can also change our password whenever in need .
7. A user can see all the events he/she has registered for.

Backend

The backend consists of mainly 2 parts to be done:

1. **Authentication**:

Our website uses inbuilt Django forms. When a user or an organization visits our website then if they are registered then they have to login into our website directly or else they have to first register then login option is available to them.

2. **Routes and Error Handling**:

Most of our project is written in JavaScript and Django.

Handling is the most important part of the server. Server basically needs to handle all the requests made by the frontend and call the database if needed.So, for coding the server we have to use Django**.** The basic requests which our server needs to handle are GET, POST, EDIT, DELETE.

Database Management

We have used SQLite database.

There are two crucial things needed to be handled in database:

1. **Creating models:**

Creating models means creating different types of classes in the database, for storing all kinds of data. We have basically created 8 models:

- Customer
- Carts
- Product
- Event Registered

2. **Integrating database to the software:**

SQLite is an in-built database present in Django, so it is already integrated in it.

3. **Uploading of Events by Organizations-**

The organization can upload the information of the upcoming events that they are hosting through the backend of our website, where they can fill the information about their events, which are the fields of the models.

4. **Removal of any Event-**

The organizations can also remove any particular event which is canceled or is already conducted by choosing remove event in the backend.

3. Integrating frontend, server, database and API

- Our frontend is written using HTML, CSS and js .
- Server is written using Django.
- Databases used are SQLite

So, we need to integrate all this. So, for that first all the requests made by frontend needs to be responded to by our server. So, for that first we have defined.
that all the files going to be rendered by our server, so in this way all us frontends are connected to the server.
Now as Django is having an integrated SQLite database thus the database and server are already connected.

4. Security Details

All the data provided by the user is stored in a database, whose access is only to the moderators, who can modify the data only following certain terms and conditions. Also, the passwords provided by the user are never stored in the database in its original form, basically what we have stores is the hash value which is created using a well secured cryptographic algorithm, which ensures that if any of the third men access our database by any means, can’t decrypt the password. Also, all the databases and servers we used were in the software zone for a long time and used by very large companies. Let's say some third-party tried to modify our database by sending requests using a postman. So, for securing from such attacks we have defined middleware so most of the pages are rendered if and only if an authenticated user is logged in, also for modifying the contents added by some particular user can only be modified by that particular user and it is done by the middleware.

Application of the project

- It works as a single platform for people who are looking for different competitive events happening around them.
- Users and organizations from around the globe can easily register themselves on Let’s Compete.

- Further, users can participate in events of their choice (by registering through our website) whereas organizations can create events on this platform for different enthusiasts.

WORKING OF THE APPLICATION

Innovation in the Project

The biggest problem of the students was that they could not access all the related information and based on them various opportunities at one place, this is what LET'S COMPETE does. It gathers all the information at one place. Our innovation is:

- Students don't have to search various websites for their academic needs.
- They would have all the required courses in one place.
- They would also have various internships, jobs and scholarships related things.
- Further we are not only focusing on academic but also cultural as well.
- It also helps in making contact with professors as they post-course and various companies for interns and jobs related things.

Thus, it proves to be a complete package for the students, professors and companies.

Future scope

In future we will make this website more user friendly so that it may reach the largest set of students possible. We might:

1. Provide the facility of webinars.
2. Make the platform more user friendly
3. Provide the facility of live chatting of students and doubt clearing
4. Enhance the facilities according to need of the consumer
5. The website will be enhanced according to the new fields emerging.

Instructions to run this project on your local server

On windows

1. Install Python in your system
2. Open the terminal of your system and go to the project directory
3. Run command on terminal: python –m pip install django
4. Run command on terminal: python –m pip install Pillow
5. Run command on terminal: python manage.py runserver
6. Open the localhost url (27.0.0.1:8000) that the terminal provides on success.

On mac ios

1. Install Python3 or latest version in your system.
2. Open the terminal of your system and go to the project directory.
3. Run command on terminal: python3 –m vnev xyz
4. Run command on terminal: source :xyz/bin/activate
5. Run command on terminal: python –m pip install django
6. Run command on terminal: python –m pip install Pillow
7. Run command on terminal: python manage.py runserver
8. Open the localhost url (27.0.0.1:8000) that the terminal provides on success.