**Author**

Name: Shaonkoli Saha

Roll No.: 21f1001196

Email ID: [21f1001196@ds.study.iitm.ac.in](mailto:21f1001196@ds.study.iitm.ac.in)

About me: I am a software professional having 9+ years of experience. My current position is Senior System Engineer in Oracle Plsql.

**Description**

This small application can collaborate sponsors and influencers on the same page.

Sponsors can create campaigns and search for perfect influencers to execute those campaigns.

Influencers can search for campaigns with their strength and can get paid.

Admin can check the activities of sponsors and influencers. Admin can control the app for avoiding frauds.

**Technologies Used**

* **SQLite database:** To store admin, influencer, sponsor and campaign details. Also use to log activities of sponsors and influencers.
* **HTML:** For GUI. This is using to take instruction from the use and display the output after processing the information. N MVC architecture, this is using as View.
* **Flask:** This is using to process the information received from user. In MVC architecture, this is using as Controller.
* **Flask SQLAlchemy:** This is using to communicate with database using python. CRUD functionality has been applied by this.
* **API:** To pass the information to display in HTML page.
* **Dictator:** After processing information, this method used to move to required path.
* **Bootstrap:** This is used for designing the web page.
* **Render\_Template:** After processing information, this method used to display the required HTML page.
* **Javascript:** Used to execute processes in front end using data provided by APIs.

**DB Schema Design**

* **User:** This table contains the user information. Table structure is as below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Purpose** |
| Id | Integer | Primary | Unique identifier of each user |
| Username | Varchar | Not null | Name of the user |
| Email | Varchar | Not null | Mail id of the user |
| Password | Varchar |  | Password of the user |
| Active | Boolean |  | If the user is active or not |
| Fs\_uniquifier | Varchar |  | Modified password with unique key |
| User\_type | Varchar |  | Type of the user |
| Last\_login | datetime |  | Last log in of the user |

* **Sponsor:** This table store the information of sponsor. Table structure is as below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Purpose** |
| Id | Integer | Primary key | Unique identifier of each sponsor |
| Name | Varchar | Not null | Name of the sponsor |
| Email | Varchar | Not null | Email of the sponsor |
| Industry | Varchar |  | Industry of the sponsor |
| Budget | Integer |  | Budget the sponsor has |
| Bio | Varchar |  | Description of the sponsor |
| flagged | Varchar |  | If the sponsor is flagged or not |

* **Influencer:** This table store the information of influencer. Table structure is as below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Purpose** |
| Id | Integer | Primary key | Unique identifier of the influencer |
| Name | Varchar | Not null | Name of the influencer |
| Email | Varchar | Not null | Email of the influencer |
| Category | Varchar |  | Category of the influencer |
| Bio | Varchar |  | Description of the influencer |
| Reach | Integer |  | Reach of the influencer |
| Flagged | Varchar |  | If the influencer is flagged or not |

* **Campaign:** This table store the information of campaigns. Table structure is as below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Purpose** |
| id | Integer | Primary key | Unique identifier of each campaign |
| Name | Varchar | Not null | Name of the campaign |
| Description | Varchar |  | Description of the campaign |
| Start\_date | Varchar |  | Start date of the campaign |
| End\_date | Varchar |  | End date of the campaign |
| Industry | Varchar |  | Industry of the campaign |
| Budget | Integer |  | Budget of the campaign |
| Visibility | Varchar |  | Visibility of the campaign |

* **Spon\_Influ\_Camp:** This table is link between campaign, sponsor and influencers. Table structure is as below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Purpose** |
| Id | Integer | Primary key | Unique identifier of each request |
| Spon\_id | Integer |  | Sponsor id of the request |
| Influ\_id | Integer |  | Influencer id of the request |
| Camp\_id | Integer |  | Campaign id of the request |
| Message | Varchar |  | Message from the influencer |
| Status | Varchar |  | Status of the request |
| Request\_from | varchar |  | Who send the request |

**Architecture and Features**

**Architecture:**

* Controller: /MAD2\_Project/app.py
* Api: / MAD2\_Project/api.py
* Templates: / MAD2\_Project/templates
* Database: / MAD2\_Project/mad-2.sqlite3

**Features Implemented:**

* Login: Admin, Sponsor and Influencer can log in through different pages.
* Admin will automatically add when the database will be created for the first time.
* Sponsor can create campaigns and request influencers.
* Influencer can see the requested campaigns and can accept or reject the request.
* Influencers can search campaigns and request sponsors.
* Sponsors can accept or reject the request send by influencers.
* Influencers and sponsors can both update their profile.
* Sponsor can update campaigns.
* Admin can see the details of sponsors and influencers.
* Admin can delete user or flag user to restrict them to use the application.
* Admin needs to approve sponsors to use the application.
* Admin can see number of campaigns completed by each influencer and sponsors.
* Admin can also see number of sponsors, influencers and campaigns are there in this platform.

**Video**

https://drive.google.com/file/d/1fs0Z3o2gNxAQXKbK5EQpTO5AxqRYUXGM/view?usp=sharing