



---

## Project Statement - MAD I

Updated automatically every 5 minutes

---

# Modern Application Development I

## Project Statement

### Influencer Engagement and Sponsorship Coordination Platform

It's a platform to connect Sponsors and Influencers so that sponsors can get their product/service advertised and influencers can get monetary benefit.

### Frameworks to be used

These are the mandatory frameworks on which the project has to be built.

- Flask for application code
- Jinja2 templates + Bootstrap for HTML generation and styling
- SQLite for data storage

**Note:** All demos should be possible on your local machine.



---

## Project Statement - MAD I

Updated automatically every 5 minutes

---

### 1. **Admin - root access**

- An admin can monitor all the users/campaigns, see all the statistics
- Ability to flag inappropriate campaigns/users

### 2. **Sponsors - a company/individual who wants to advertise their product/service**

- Sponsors will create campaigns, search for influencers and send ad requests for a particular campaign.
- Sponsors can create multiple campaigns and track each individual campaign.
- They can accept ad requests by influencers for public campaigns.
- Each Sponsor may have;
  - a. Company Name / Individual Name
  - b. Industry
  - c. Budget

### 3. **Influencers - an individual who has significant social media following**

- An influencer will receive ad requests, accept or reject ad requests, negotiate terms and resend modified ad requests back to sponsors.
- They can search for ongoing campaigns (which are public), according to category, budget etc. and accept the request.



## Project Statement - MAD I

Updated automatically every 5 minutes

- a. Name
- b. Category
- c. Niche
- d. Reach (can be calculated by number of followers / activity etc.)

**Terminologies**

**Ad request** : A contract between campaign and influencer, stating the requirements of the particular advertisement (E.g. show Samsung s23 in 3 videos for 10 seconds each), the amount to be paid etc.

Ad request may have:

- a. campaign\_id (Foreign Key to Campaign table)
- b. influencer\_id (Foreign Key to Influencer/user table)
- c. messages
- d. requirements
- e. payment\_amount
- f. status (Pending, Accepted, Rejected)

**Campaign** : A container for ads requests for a particular goal (E.g. advertisement for Samsung s23). It can have multiple Ad requests, a campaign description, budget, ability to set public or private

Campaigns may have:

- a. name
- b. description
- c. start\_date
- d. end\_date
- e. budget
- f. visibility (public, private)
- g. goals



## Project Statement - MAD I

Updated automatically every 5 minutes

only to get the flow of the application and what should appear when a specific user navigates from one page to another. It is NOT mandatory to exactly replicate the views given in the wireframe. Students can work on their own frontend idea.

### Core Functionalities

#### 1 Admin login and user login

- A login/register form with fields like username, password etc. for sponsor, influencer and admin login
- You can create separate forms for each type of user
- You can either use a proper login framework, or just use a simple HTML form with username and password (we are not concerned with how secure the login or the app is)
- The app must have a suitable model to store and differentiate all the types of user of the app.

#### 2. Admin Dashboard - for the Admin

- The application must have an admin dashboard which display all the relevant statistics of the application, e.g. active users, campaigns (public/private), ad requests and their status, flagged sponsors/influencers etc.
- Students can decide what more statistics to be shown apart from the ones given above



## Project Statement - MAD I

Updated automatically every 5 minutes

campaign - e.g. start\_date, end\_date, budget and/or other fields

- Delete an existing campaign

#### 4. Ad request Management - for the sponsors

- Create ad requests based on the goals on the campaign
- Edit an existing ad request - e.g. influencer\_id, requirements, payment\_amount, status
- Delete an existing ad request.

#### 5. Search for influencers, public campaigns

- The sponsors should be able to search for relevant influencers based on their niche, reach, followers etc.
- The Influencers should be able to search for public campaigns based on their niche, relevance etc.

#### 6. Take action on a particular ad request - for the Influencers

- Ability to view all the ad requests from all the campaigns
- Ability to accept/reject a particular ad request
- Ability to negotiate the "payment\_amount" for a particular ad

### Recommended Functionalities

- API resources created to interact with the users, ad requests and/or



## Project Statement - MAD I

Updated automatically every 5 minutes

- APIs can either be created by returning JSON from a controller or using flask extension like flask\_restful
- External APIs/libraries for creating charts, e.g. ChartJS
- Implementing frontend validation on all the form fields using HTML5 form validation or JavaScript
- Implementing backend validation within the controllers of your app.

### Optional Functionalities

- Provide styling and aesthetics to your application by creating a beautiful and responsive frontend using simple CSS or Bootstrap
- Incorporate a proper login system to prevent unauthorized access to the app using flask extensions like flask\_login, flask\_security etc.
- Implement a dummy payment portal (just a view taking payment details from sponsors for an ad request)
- Any additional feature you feel is appropriate for the application

### Evaluation



## Project Statement - MAD I

Updated automatically every 5 minutes

- The report must include the following things;
  - Student details
  - Project details, including the question statement and how you approached the problem statement
  - Frameworks and libraries used
  - ER diagram of your database, including all the tables and their relations
  - API resource endpoints (if any)
  - Drive link of the presentation video
- The project report must be included as a PDF **inside** the root submission folder and NOT along with it.
- All code to be submitted on portal in a single zip file (zipping instructions are given in project document - Project Doc T22024)
- Students have to create a brief (3–5 minute) video explaining how you approached the problem, what you have implemented, and any extra features
- The video must be uploaded on the student drive with **access to anyone with link** and the link must be included in the report
  - This will be viewed during or before the viva, so should be a clear explanation of your work
- Viva: after the video explanation, you are required to give a demo of



Published using Google Docs

[Report abuse](#)[Learn more](#)

## Project Statement - MAD I

Updated automatically every 5 minutes

- Other questions that may be unrelated to the project itself but are relevant for the course

### Instructions

- This is a live document and will be updated with more details (wireframe)
- We will freeze the problem statement on or before 19th May 2024, beyond which any modifications to the statement will be communicated via proper announcements.
- The project has to be submitted as a single zip file.