# Mordern Application development project report

* Student details:
* **Name: Tushti Goyal**
* **Roll no:** **21f3003054**
* **Program: BS Degree in Data Science and Applications**
* **Level: DIPLOMA**
* Project details, including the question statement and how you approached the problem statement
* **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.**

I inferred my project having three roles and three interactive objects. I tried to approach it by creating a database based on the core functionality and starting the implementation, directly on the controllers, but that was not successful.

Then, I created an MVC structure with the db.Model classes being a façade layer between the Functional classes and database. Only db.Model is used for retrieving the and adding the data on the database. My classes.py file, consists of all the functional classes with the logic needed for functions in that. Then the controller file handles the app routes and app.py finally creates the file. There is a static and templates folder for the View component of my structure.

Functionality:

Admin, Influencer and Sponsor being the three roles, and Campaigns, Ads and requests being the three objects. I have successfully implemented the login page, the three roles and objects with the request object “travelling” between the Sponsor and influencer and the flag functionality with and unflag functionality.

Only a sponsor can create a campaign after which they can create a ad. After creating an add, requests can be sent by looking up all the influencers on the system to multiple or one influencer at the same time with a different message to each.

Influencer will view the request and can negotiate the payment amount with a custom message to negotiate the terms. After which the request is sent back to the Sponsor where they can negotiate further. A request can be rejected or accepted at any point in time in the negotiations.

When a sponsor is deleted, all its campaigns are deleted. When a campaign is deleted all its ads are deleted. When an ad is deleted all its requests are deleted. Flagged and unflagged is not implemented this way yet. If a flagged user tries to login, the app asks them to contact the admin.

I am trying to implement the stats screen, but there is trouble in the savingfig in the static folder.

* Frameworks and libraries used: Bootstrap 5-modal view ,JS: JQuery (non slim), jsdeliver – To handle posts requests on click , Awesome-font- provides icons, CSS-some manual styling, Flask- to define app routes, create the application and the models , datetime, SQLAlchemy-Database connection and management, matplotlib.
* ER diagram of your database, including all the tables and their relations
* Drive link of the presentation video