**Institute Inbound Media Hub**

(DAC KP Feb 2020)

**Students’ Details:**

|  |  |
| --- | --- |
| 200250120084 | Sanket Jadhav |
| 200250120068 | Pramod Khodade |
| 200250120036 | Jamdar Amrut |
| 200250120016 | Bhagyashri Patil |

**Project Objective**

* In this application members of an institute will be able to create an account by registering through sign-up.
* A valid user will be able to share the information (create a post).
* Information can be anything such as some technical information, notice, scheduled events or other.
* Post can be in the form of text and image (1 image per post).
* Every valid user can see all the posts that is from themseleves and other users.
* Every valid user can vote the post as positive and negative (upvote and downvote) according to the usefulness of the post.
* Every valid user can comment on the post to express their opinion on the same.

**Work flow**

* Signup
* In a signup module, all the user details are taken from end user with validation to the input boxes, all the detailes are sent to signup service.
* Then those details are stored in a database.
* Email id of user trying to register is fetched and verification token is sent to the same.
* If user accessed the verification link, then it is marked as valid user and allowed to log-in to application (enabledUser is set to true).
* A user is invalid user untill and unless verification is completed.
* **Create Post**
* A post can be created only when a user is logged-in to application.
* A post object is created, which contains post title, url, type of post, image and description.
* All the entered details are then passed to database.
* Each post has its own id created and stored in database.
* Whenever user clicks view post these details are fetched and displayed on the page.
* Other users then only can view and react to the post, but won’t be able to edit it.

**Problems we faced and solved.**

* In first sprint,
* In a signup process we wanted to perform some validation to every user who want to register.
* But couldn’t process with it.
* When referencing some websites we came to know that a  ***mail sender*** interface can be used to solve this issue.
* We tried it in a simple mail sending application first.
* We got various exceptions, again we find the solution by referring the spring documents.
* When we succeded to send a mail with verification link we implied it in our application.
* In second sprint,
* When we were trying to store and fetch image in database we were not able to do it.
* Then we use form data to store image, we append the image to form data.
* Then we retrived the bytes of that image object and stored it in database.
* While fetching bytes are fetched from database then these bytes are again converted to image to display.
* This gave us the desired output.

**Learnings from project.**

* We worked as a team, and shared the responsibilities among us, we helped and co-ordinated each other at every phase.
* We learnt to use entire three tier arhitecture of application development.
* What we learnt in angular, spring and sql, we practically implied it here.
* We tried to learn and apply new concept.
* We learnt to find a solution to overcome the problems we face while developing application.