**Name of project: Kickstarter Application**

**Problem Statement**

People are unable to promote their ideas to public due to lack of funding. Due to this, such great ideas do not become a reality.

**Approach**

Developed an application to support crowdsourcing for projects where users can convey their project ideas to viewers and get funding for their projects which they cannot afford on their own.

In return, the customers would get services based on the amount they have contributed.

Created different workspaces relative to different roles like admin, creator, donor and guest.

**Getting started**

The project is deployed in two applications: Angular JS and Rest.

Project names in Eclipse: ‘Project’ and ‘AngularRESTApplication’

**Note: Please open the application in Incognito mode to avoid any issues due to cookies.**

**Motivation for the project**

The idea behind this project was to develop a web application by making the use of technologies like Angular JS, REST, Spring, Hibernate and MySQL together.

**Architecture**

Angular (View , Ctrl) ⇔ REST ⇔ Service ⇔ DAO ⇔ Hibernate.

### Angular JS (Front End)

### REST (Rest handles any request using Model and sends response (JSON))

### Service (Business Logic for the application- Converting Model to Entity and reverse)

### DAO (Database Access performing CRUD)

### Spring Injection for (SessionFactory/Controller/Service/DAO)

### Hibernate (Hibernate Mapping: hbm.xml / annotation)

**Prerequisites:**

Software’s required: Eclipse with Apache Tomcat server, MySQL Workbench.

Note: Please import the SQL file database\_mrunalini.sql attached in MySQL Workbench.

**Description in detail:**

* Login page (Authentication - using username & password).
* Password is hashed in database.
* A guest can Browse Projects based on category and view details without Login.
* Registration Form with basic validation. (For Creator & Donor)
* 3 Different Dashboard (Workspace) after login.
  + 1. Admin
    2. Creator
    3. Donor
* Workflow:

Admin (Needs Authentication – A query is attached in the document Admin\_Query.txt to create the credentials for admin).

A) Create categories based on the type of application.

Validation done in such a way that the categories should be Unique.

B) Delete any spam projects created.

Should mention the reason for deleting which would be stored in database. The status of the project is changed to inactive in database.

C) Delete/Disable Categories based on presence of PROJECT.

* + - 1. If No Project are present under a category, admin can delete.
      2. If Projects are presents under a category, admin can disable.

Creator: (Needs Authentication)

1. Creator needs to be Registered in the system before creating a project.

Registration details:

Name (First name, last name)

UserName

Password

Contact: Email ID and Phone Number

Role – (Creator or Donor)

1. Creator creates a project
   * + 1. based on category
       2. create a title
       3. description.
       4. total amount they would like to raise and
       5. duration of projects.
2. Each Projects needs to have at services. Service can be defined as:

Define amount for which they will receive the product at earlier date (Example: If you pay 500$ you would receive 1 product or if you pay 900$ you would receive 2 products)

1. After Project duration is over; Services are disabled. But project should be visible.

Donor:

Without Authentication

* + 1. Browse All projects based on category.
    2. View Project Details along with services.

Need Authentication.

1. Select a single service for Payment based on amount of service.
2. Payment capture
   * + 1. Donor Details
       2. Credit Card Info.
       3. Payment Date

**Database Design**

For ER diagram, refer ER\_diagram.mwb attached.

Screenshot for the same:

