A PROJECT REPORT

ON

ECOSYSTEM

By

Sooraj Mishra (1503310231)

Shubham Kumar Jha (1503310220)

Saransh Gupta(1503310190)

Sanuj Kumar(1503310189)

Department of Computer Science & Engineering

Raj Kumar Goel Institute of Technology, Ghaziabad

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

May,2019

ECOSYSTEM

By

Sooraj Mishra(1503310231)

Shubham Kumar Jha(1503310220)

Saransh Gupta(1503310190)

Sanuj Kumar(1503310189)

Submitted to the Department of Computer Science & Engineering

in the partial fulfilment of the requirements for the degree of

Bachelor of Technology

Raj Kumar Goel Institute of Technology, Ghaziabad
Dr. A.P.J. Abdul Kalam Technical University, Lucknow
May,2019

CERTIFICATE

This is to certify that Project Report entitled “eCOSystem” which is submitted by Sooraj Mishra, Shubham Kumar Jha, Saransh Gupta, and Sanuj Kumar in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, is a record of the candidate own work carried out by him under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date:

Project Guide Name & Signature
(Mr. Sandeep Vishwakarma)

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

SANUJ KUMAR (1503310189)

Signature

SARANSH GUPTA (1503310190)

Signature

SHUBHAM KUMAR JHA (1503310220)

Signature

SOORAJ MISHRA (1503310231)

Approved By:

Signature
(Mr. Sandeep Vishwakarma)
Project Coordinator
CSE Department
RKGIT, Ghaziabad

Signature
Dr. Sachi Gupta
HOD
CSE Department
RKGIT, Ghaziabad

ACKNOWLEDGMENT

It gives us a great sense of pleasure to present the report of the B. Tech. project undertaken during B. Tech. final year. We owe special debt of gratitude to Assistant Professor Mr. Sandeep Vishwakarma, department of Computer Science & Engineering, Raj Kumar Goel Institute Technology, Ghaziabad for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance, have been a constant source of inspiration for us.

We also take the opportunity to acknowledge the contribution of Dr. Sachi Gupta Head & Professor, Department of CSE RKGIT Ghaziabad for her support and assistance during the development of the project.

Signature

SANUJ KUMAR (1503310189)

Signature

SARANSH GUPTA (1503310190)

Signature

SHUBHAM KUMAR JHA (1503310220)

Signature

SOORAJ MISHRA (1503310231)

ABSTRACT

The proliferation of candidates pursuing engineering has embarked the cycle of innovation, technology and knowledge. But still, out of all, only a hand few of candidates get to acknowledge its true meaning. Progression and growth in the right direction is paramount when it comes to achieving success and defining a path for your goal. The growth of social networking has established a well-defined platform for social interaction and relationship between individuals, offering a gigantic pool of information and knowledge relevant to the specific set of individuals, thereby fostering the growth of everyone as a unit i.e. the experiences earned, the hassles confronted and the paths taken is there for all to acknowledge and learn so as to further the line of progression.

The system features a network of individuals connected to one another as nodes and the links between them demonstrates their relationship. The profile of any candidate grows as more activities and experiences are added to it. A candidate can guide or follow anyone on the basis of skill-set and common interest so as to get a clear picture of the dos and don'ts. Candidates with accounts on sites like LinkedIn, Facebook, GitHub etc. can link it to their profile for easier communication.

Our project will also provide the platform where student will get the benefits to be interacted with the industry professionals. In this project a social networking site exclusive for the college has been created whereby student, faculty, selective people in the industry would be able to socialize and share their knowledge. This paper describes the features implemented in this project and presents a glimpse of how the projects work.

Our intention to build this system is mainly focused on introducing a culture and an idea of ubiquitous development where everyone has a chance to learn and follow everyone else there by enabling any individual to share his/her experiences, the paths they took, the methods they followed and the goals they have, for all to see and grasp something out of it. Though the best solution approach will always remain open to interpretation but our work has contributed in developing a familiar approach which is more much reachable to everyone and provides motivation for future work in this area.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTROUCTION

eCOSystem is a platform whose main purpose is to introduce a culture of social interaction and ubiquitous learning process for everyone.

This site lets people find working associates, students, colleagues whom they already know. You “connect” with them through the site, and they become part of your network.

Once you have connected with a person, you will have access to their list of connections, their work experiences, their area of interest, their technical skills and they have done in their studies.

There are several features provided in this project, such as increase your connections or networks through your profile and also get the link of networks in your interested area. If you are confused with what technologies you should learn that will beneficial in future. You can check the profile of other person whom you know and make a connection with him and ask him for your career counselling.

1.1 Objective and Scope

We believe that the impact of organizational knowledge is an important aspect to study, since it involves all international companies and it is an aspect that all managers should consider carefully. Knowledge transfer is always complex, but the complexity increases even further when a transnational aspect is added. Furthermore, effective and good communication between parties is an important aspect for knowledge transfer. It can, if managed properly, reduce both risks and costs since both money and time can be saved when misinterpretations and misunderstandings are avoided. The choice of appropriate transmission channels and the effective matching process are a central part of the communication process.

Sometimes it is not always possible for everyone to take right decision which will beneficial to them to build their career. There might be lot of confusion at the time in which everyone has to choose the right path for their achievement of goal and the main reason is communication gap, less knowledge to that field and not proper preparation.

This social network service on the building and verifying of online social network for communities of learner and experienced one, who share their interests and activities or who are interested in exploring the interests and activities of others, and which necessitates the use of software. Most services are primarily web based and provide a collection of various ways for user to interact, such as queries, Advices and file uploading.

So building this project will help to those individuals who pursue their dreams by getting proper guidance and direction for their field of area of interest. Our aim is to provide a platform through which individuals will connect with each other and share their experiences in the same field of interest.

The best solution approach to this problem will remain open to interpretation for several years but we could at least offer a platform for providing a better channel for communication and knowledge.

- Acquiring industrial knowledge is not an option for a fresher but having some idea related to the current scenario and requirements of an industry/corporate is important. Only a candidate with some experience can list out the exact details for all the other candidates. This system is built with the idea for make such details reachable to all those require so they can prepare themselves.

- Knowing what to learn at the right time and what you should learn for furthering your career in a definite stream is very important. Through this system we can be guided by many individuals with similar interest who have already been through this stage and will provide directions in a better way so that we can make the right choice.

- Everyone wants to leave behind legacy for everyone else to acknowledge so as to share their experiences, the challenges faced and their decisions so as leave a mark behind for others to know and learn. This is a vital feature of our system.

1.2 EXISTING STANDARDS

- **Take right decision for your goal**

Joining the professional medium to connect with each other will help to take right decision.

- **Step-by-Step Guides**

Our easy to use social medium wizard will help you craft the perfect goal. You have also complete control on it.

- **Start Fresh or Instantly Import Your Resume to the Web**

No other online resume builder gets you started this fast. Resume Maker allows you to start from scratch with a new resume or import your existing resume from Microsoft Word, Adobe PDF, HTML, or text file.

- **Insider Advice, Tips & Examples**

As you build your resume online, you can access important advice on how to improve each section and then you can adjust proper focus on required one.

- **Choose from Several Types of fields**

eCOSystem will help to choose your guide as per your area of interest in any field. e.g. 'for preparation of Gate Exam you can choose the option of Gate section'.

- **Make Your Resume Stand Out with Powerful Phrases**

Choose from over 150,000 professionally written career specific phrases to communicate your strengths and get the interview.

CHAPTER 2

SYSTEM OVERVIEW

2.1 System Architecture

eCOSystem is a platform whose main purpose is to introduce a culture of social interaction and ubiquitous learning process for everyone.

Once you have connected with a person, you will have access to their list of connections, their work experiences, their area of interest, their technical skills and they have done in their studies.

If you are confused with what technologies you should learn that will beneficial in future. You can check the profile of other person whom you know and make a connection with him and ask him for your career counselling.

Technology used

2.1.1 Frontend

- HTML
- CSS
- Bootstrap
- Java Script
- JQuery

HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

JAVASCRIPT

JavaScript often abbreviated as JS, is a high-level, interpreted programming language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.[9] JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it,[10] and major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms Vanilla JavaScript and Vanilla JS refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design. JavaScript was influenced by programming languages such as Self and Scheme.

BOOTSTRAP

Bootstrap 5 is the upcoming major version of the framework.

Major changes include:

- Switch from jQuery library to native JavaScript
- Dropping support for IE10
- Moving testing infrastructure from QUnit to Jasmine

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The end result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it.

JQUERY

JQuery, at its core, is a Document Object Model (DOM) manipulation library. The DOM is a tree-structure representation of all the elements of a Web page. JQuery simplifies the syntax for finding, selecting, and manipulating these DOM elements. For example, JQuery can be used for finding an element in the document with a certain property (e.g. all elements with an h1 tag),

changing one or more of its attributes (e.g. color, visibility), or making it respond to an event (e.g. a mouse click).

JQuery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event callback function definition are done in a single step in a single location in the code. JQuery also aims to incorporate other highly used JavaScript functionality (e.g. fade ins and fade outs when hiding elements, animations by manipulating CSS properties).

The principles of developing with JQuery are:

Separation of JavaScript and HTML: The JQuery library provides simple syntax for adding event handlers to the DOM using JavaScript, rather than adding HTML event attributes to call JavaScript functions. Thus, it encourages developers to completely separate JavaScript code from HTML markup.

Brevity and clarity: JQuery promotes brevity and clarity with features like "chainable" functions and shorthand function names.

Elimination of cross-browser incompatibilities: The JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another. Like other JavaScript toolkits, JQuery handles all these cross-browser inconsistencies and provides a consistent interface that works across different browsers.

Extensibility: New events, elements, and methods can be easily added and then reused as a plugin.

JQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. Web analysis (from 2017) indicates that it is the most widely deployed JavaScript library by a large margin.

JQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. JQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the JQuery library allows the creation of powerful dynamic web pages and Web applications.

The set of JQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API.

2.1.2 Backend

Technology used

- Django 2.1.7
- Recombee API
- MongoDB

Django 2.1.7

Django is a free and open source web application framework written in Python. A framework is nothing more than a collection of modules that make development easier. They are grouped together, and allow you to create applications or websites from an existing source. This is how websites even simple ones designed by a single person - can still include advanced functionality like authentication support, management and admin panels, contact forms, comment boxes, file upload support, and more. In other words, if you were creating a website from scratch you would need to develop these components yourself. By using a framework instead, these components are already built, you just need to configure them properly to match your site.

Django as "a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel."

Django offers a big collection of modules which you can use in your own projects. Primarily, frameworks exist to save developers a lot of wasted time and headaches and Django is no different.

Library used in Django

Django-social-auth

Django-crispy-forms

Django pillow

Django faker

Django argon2

RECOMBEE API

The recommendation domain consists of three principal components:

- Items
- Users
- User-Item Interactions

We assume that there is a catalog of **items** that are available to **users**. Users may **interact** with the items:

- View details about them (if there is an information page for each item present in your system).
- Rate them (explicit rating submitted by users, may also be negative),
- Purchase them (~ read the whole article, view most of a video ... depending on your domain),
- Bookmark them (add them to a wish list, or mark them as favorites),
- View some portions of them,
- or Add them to cart.

Besides their ID, the items and users may have certain properties. The **properties** are analyzed by the recommender to further improve the recommendations. Also, you may use the properties to filter or boost recommendations of some specific items. The set of properties is managed by the API user, who may define properties according to the target domain. The properties may be thought as columns in relational database table: they are of certain types and they may be of specific values for different items.

How to integrate

Integration consist of three principal components:

- Sending interactions between items and users to Recombee
- Synchronizing items catalog to Recombee
- Getting recommendations from Recombee

It can be done in multiple ways:

1. **Server side integration**

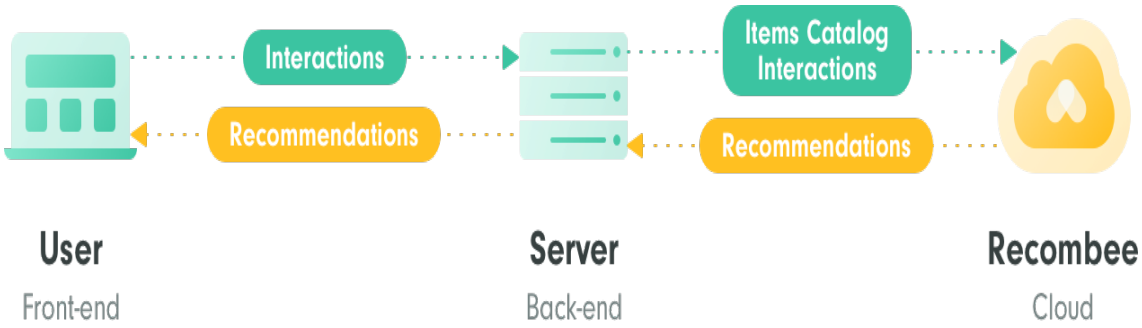


Fig 2.1: Server-Side Integration

2. **Client (JavaScript) & Server side integration**

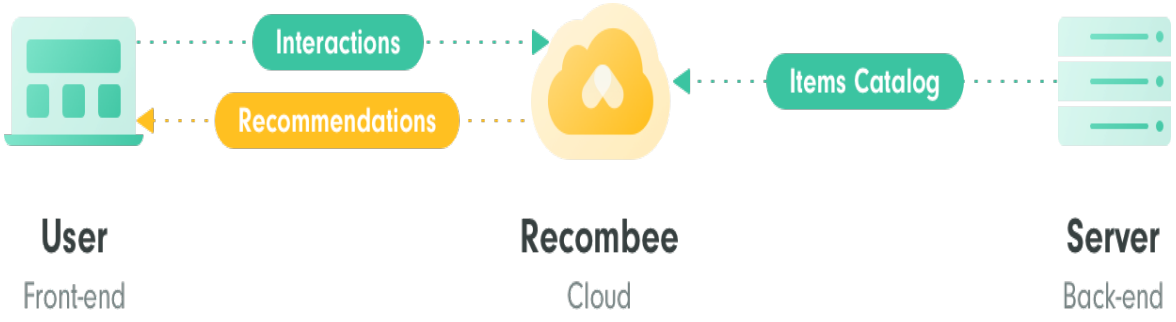


Fig 2.2: Client & Server-Side Integration

3. **Client (JavaScript) side integration & Product feed**

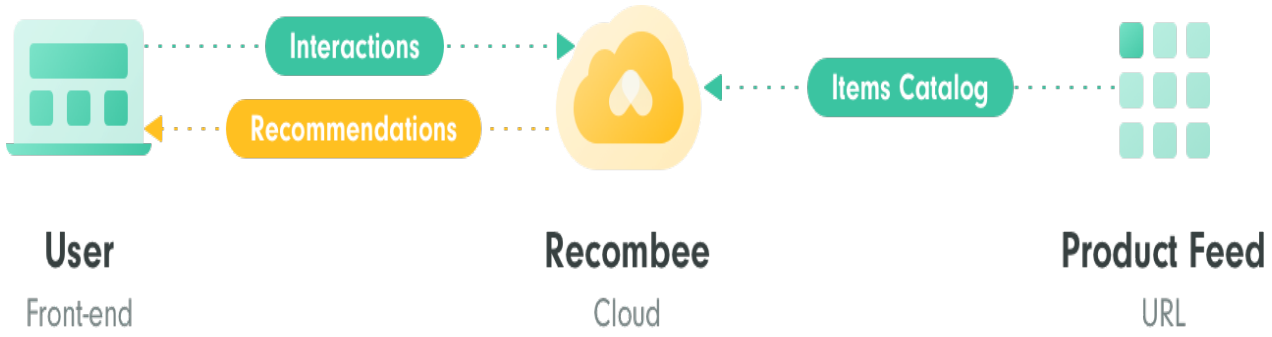


Fig 2.3: Client-Side Integration & Product Feed

MongoDB

MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.

The document model maps to the objects in your application code, making data easy to work with. Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyze your data. MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use.

FEATURE

Ad hoc queries

MongoDB supports field, range query, and regular expression searches.[8] Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size.

Indexing

Fields in a MongoDB document can be indexed with primary and secondary indices.

Replication

MongoDB provides high availability with replica sets.[9] A replica set consists of two or more copies of the data. Each replica set member may act in the role of primary or secondary replica at

any time. All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Secondaries can optionally serve read operations, but that data is only eventually consistent by default.

Load balancing

MongoDB scales horizontally using sharing.[10] The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more replicas.). Alternatively, the shard key can be hashed to map to a shard – enabling an even data distribution.

MongoDB can run over multiple servers, balancing the load or duplicating data to keep the system up and running in case of hardware failure.

File storage

MongoDB can be used as a file system, called GridFS, with load balancing and data replication features over multiple machines for storing files.

This function, called grid file system, is included with MongoDB drivers. MongoDB exposes functions for file manipulation and content to developers. GridFS can be accessed using mongofiles utility or plugins for Nginx and lighttpd GridFS divides a file into parts, or chunks, and stores each of those chunks as a separate document.

Aggregation

MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function, and single-purpose aggregation methods.

Map-reduce can be used for batch processing of data and aggregation operations. But according to \MongoDB's documentation, the Aggregation Pipeline provides better performance for most aggregation operations.

The aggregation framework enables users to obtain the kind of results for which the SQL GROUP BY clause is used. Aggregation operators can be strung together to form a pipeline – analogous to Unix pipes. The aggregation framework includes the $lookup operator which can join documents from multiple documents, as well as statistical operators such as standard deviation.

Server-side JavaScript execution

JavaScript can be used in queries, aggregation functions (such as MapReduce), and sent directly to the database to be executed.

Capped collections

MongoDB supports fixed-size collections called capped collections. This type of collection maintains insertion order and, once the specified size has been reached, behaves like a circular queue.

Transactions

Support for multi-document ACID transactions was added to MongoDB with the General Availability of the 4.0 release in June 2018.

CHAPTER 3

DESIGN

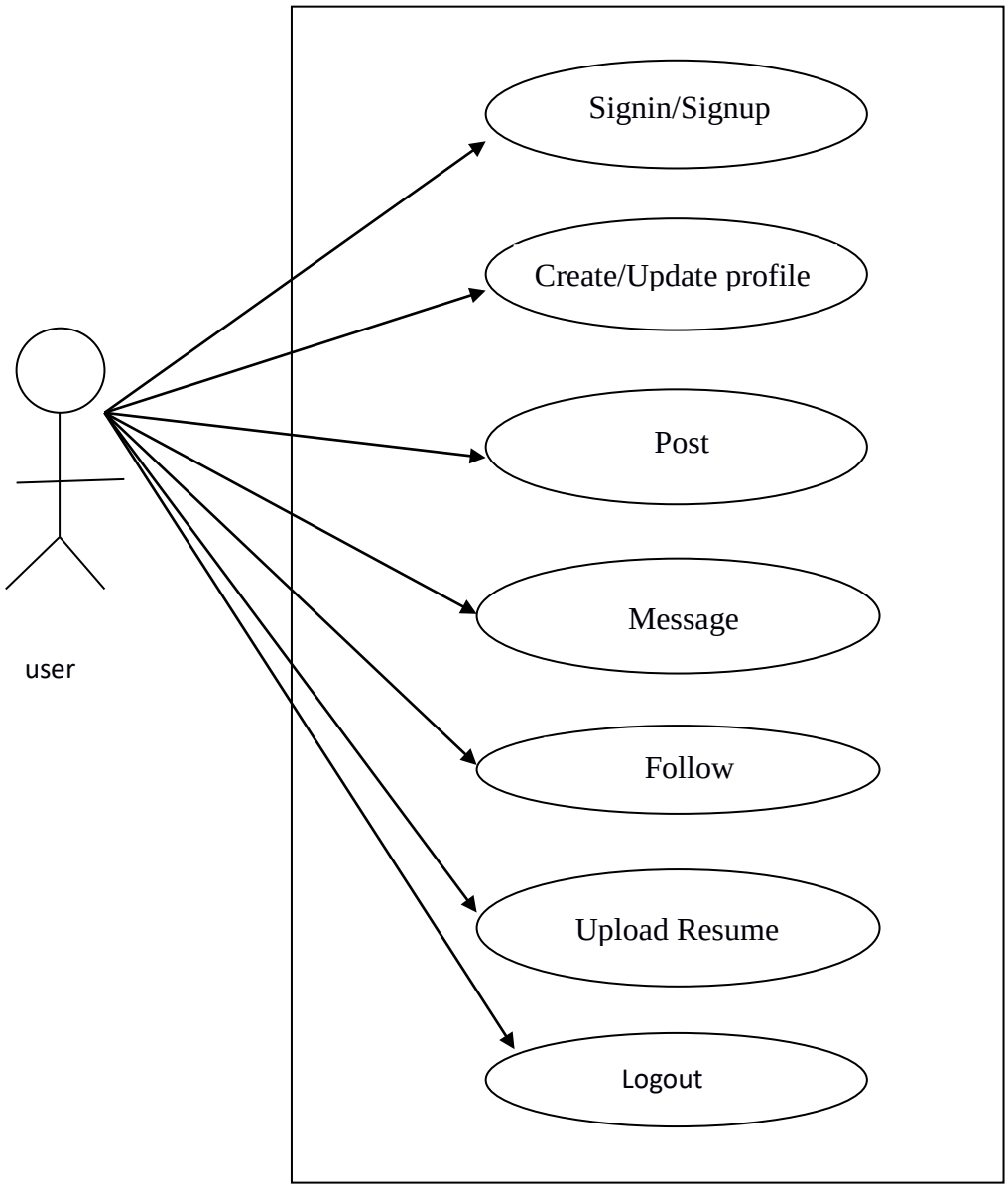


Fig. 3.1 Use case diagram

Class Diagram of Modules

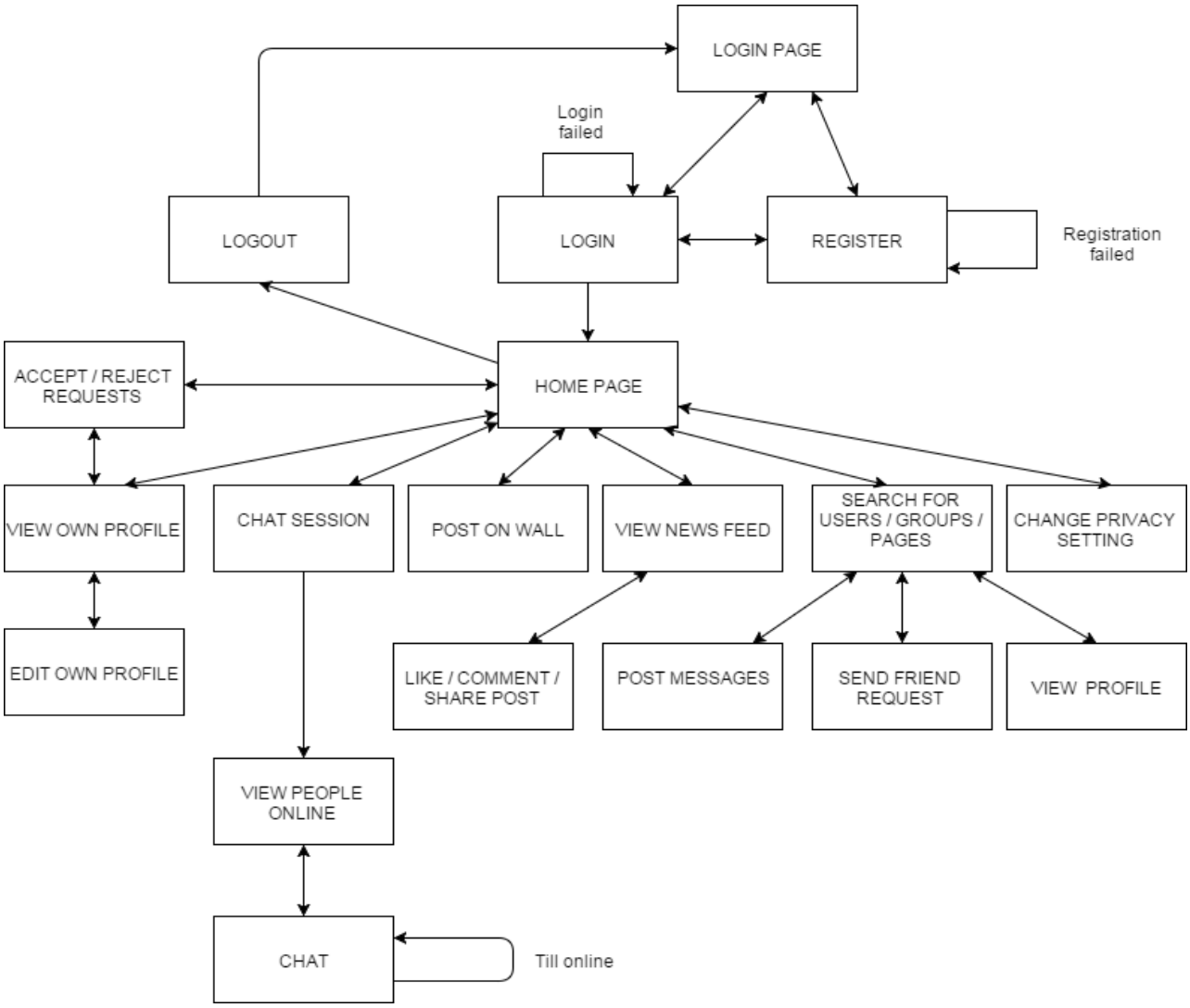


Fig. 3.2 Class Diagram

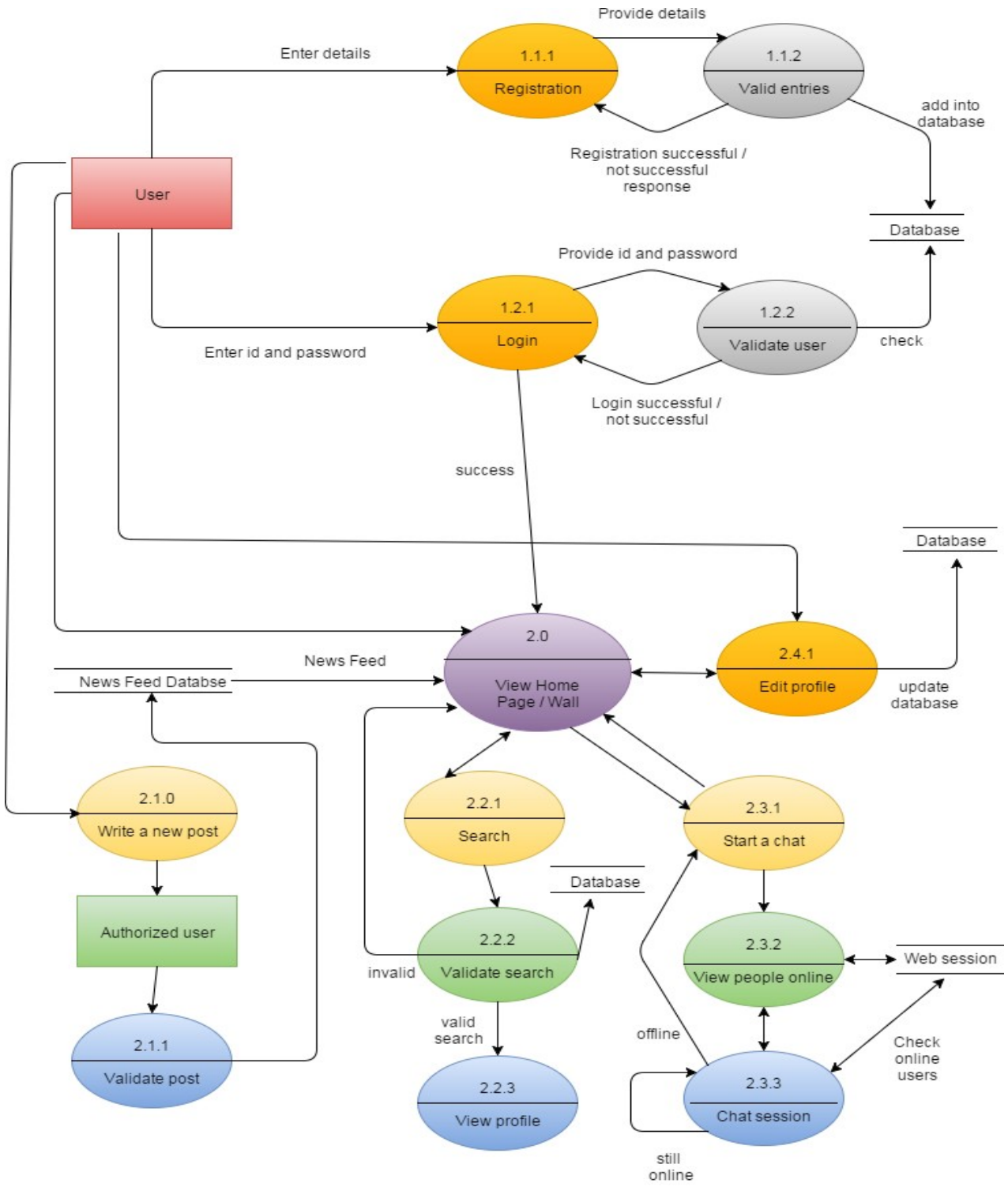


Fig. 3.3 Level 1 Data Flow Diagram

CHAPTER 4

MODULES

This system consists of various models which are the following

4.1 Authentication Module

Signup

In signup module user can register to the eCOSystem with email and creating the password.

\

Fig 4.1 Signup page

Signin:

- User can sign In to the eCOSystem by entering his email and password.
- User can also sign in by using other social accounts e.g. Google, Twitter.
- User can also change the password and if he forgets the password, he can recreate the password by clicking forgot password link.
- eCOSytem also provides security to user's password. Here password is saved in the form of hash function. Hence no one can read the password very easily.

Fig 4.2 Signin page

Logout:

User can logout of the ecosystem by clicking logout link. It enables the user to save his account form unfair means.

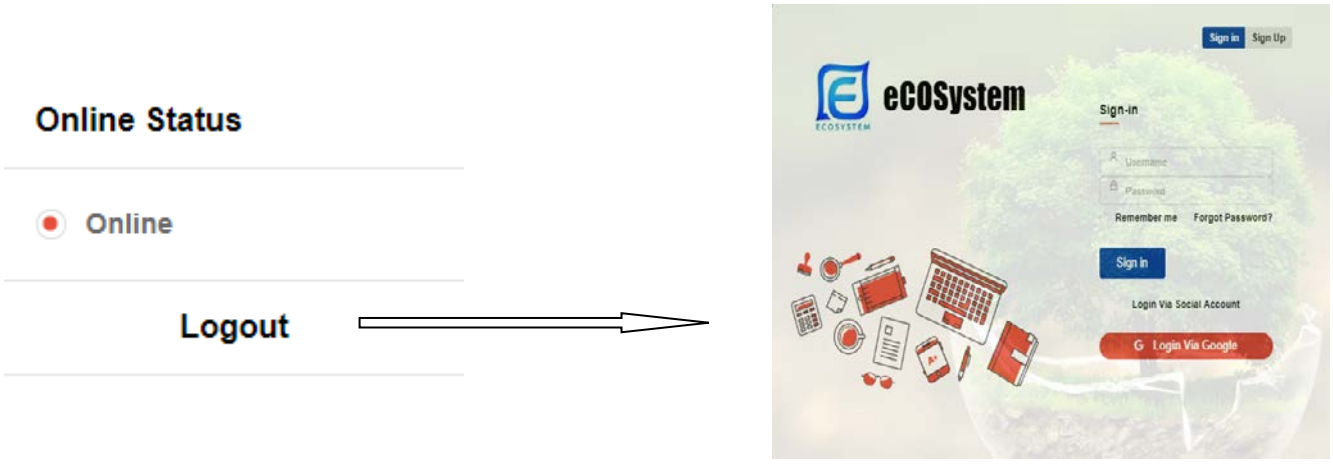


Fig 4.3 Logout

Social Login:

Login with your social (Google) account without signing up.

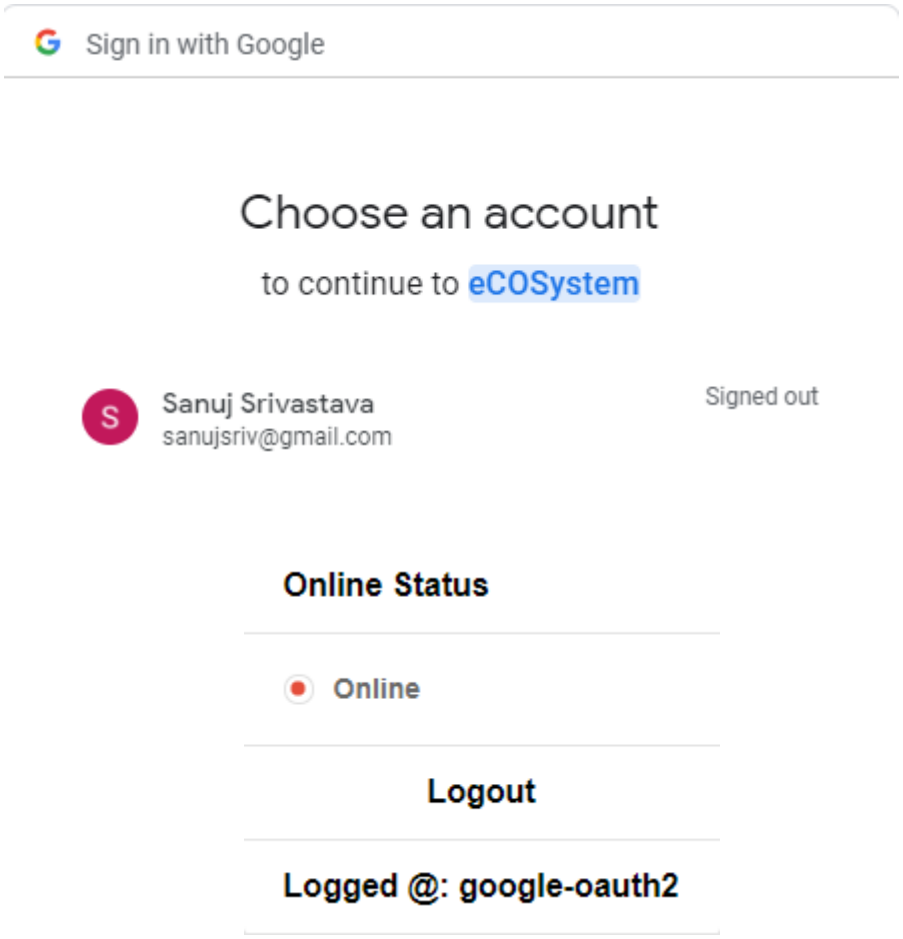


Fig 4.4 Social Logout

4.2 Session Management

eCOSystem also provides the session management feature.

4.3 Post/Feed Module

User can upload a post or any information to the eCOSystem and this post, information will be visible to all other eCOSystem user who are following him.

User can also comment and reply to the post of other users.

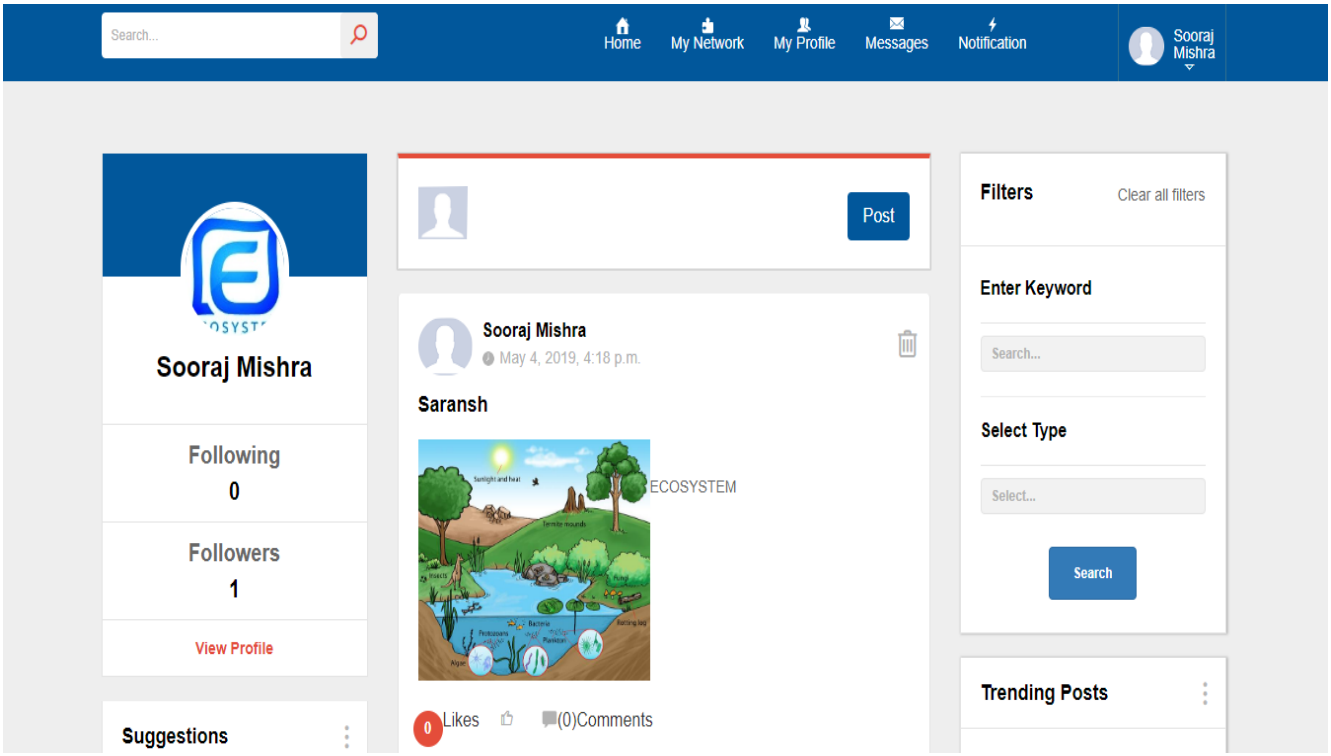


Fig 4.5 Index (PostListView)

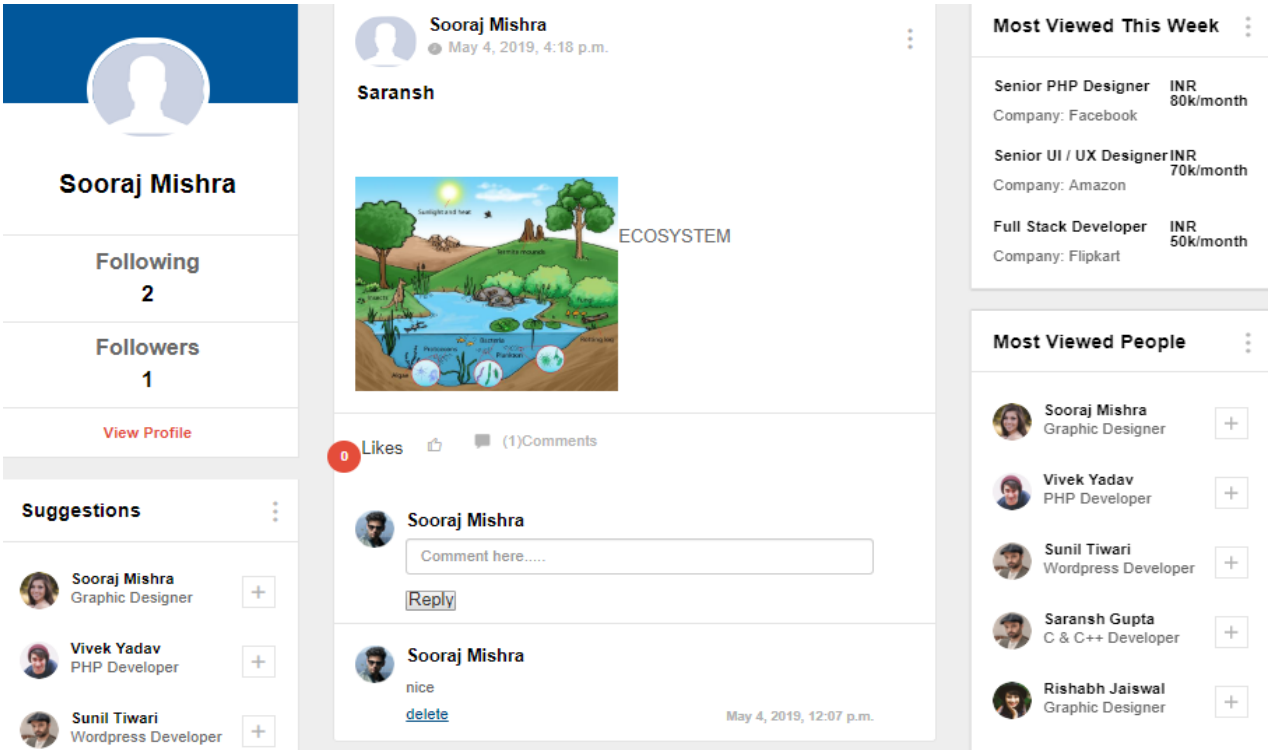


Fig 4.6 Index (PostCommentView)

4.4 Follower/Following Module

This feature helps to follow the faculty/industry professionals/alumni of the organization in the selected area of interest and after this followed people can share their experience or activities.

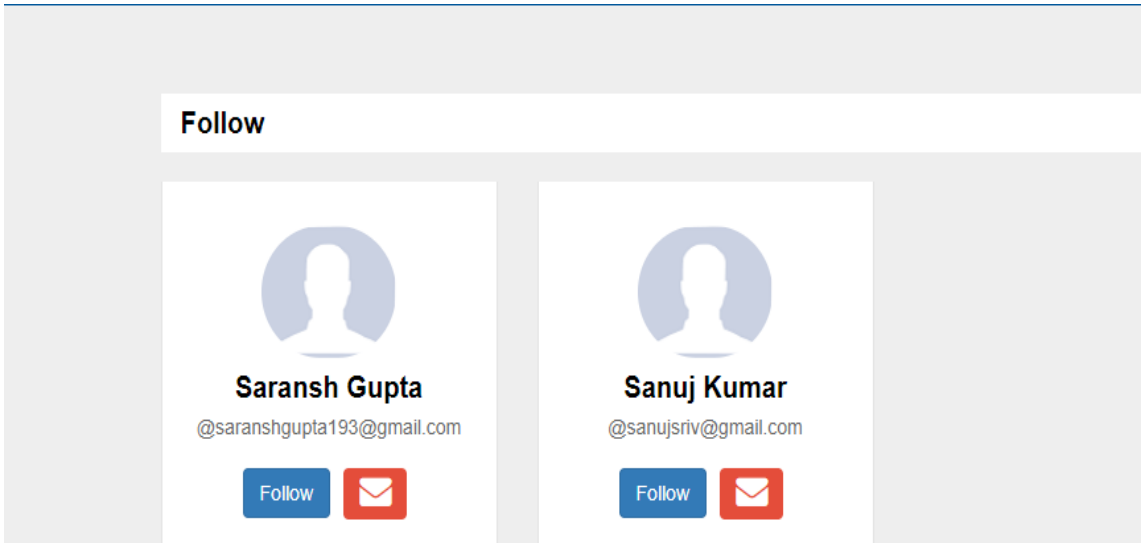


Fig 4.7 Follow

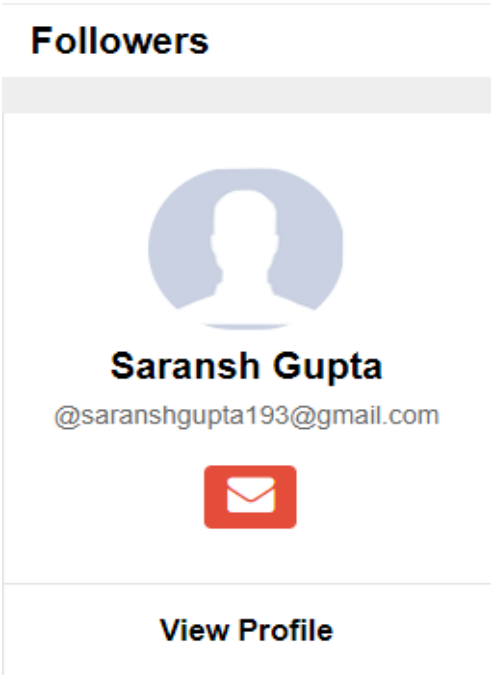


Fig 4.8 Followers

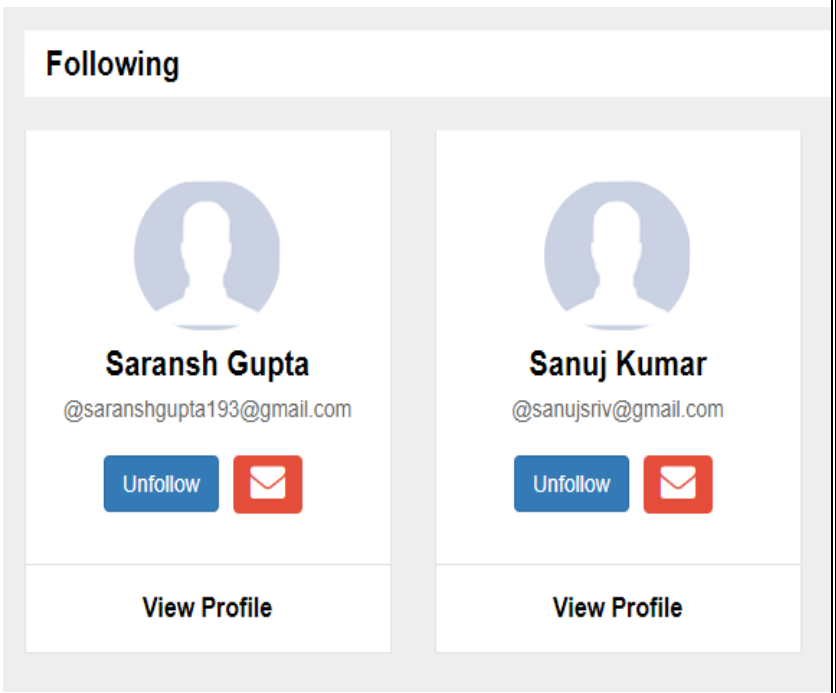
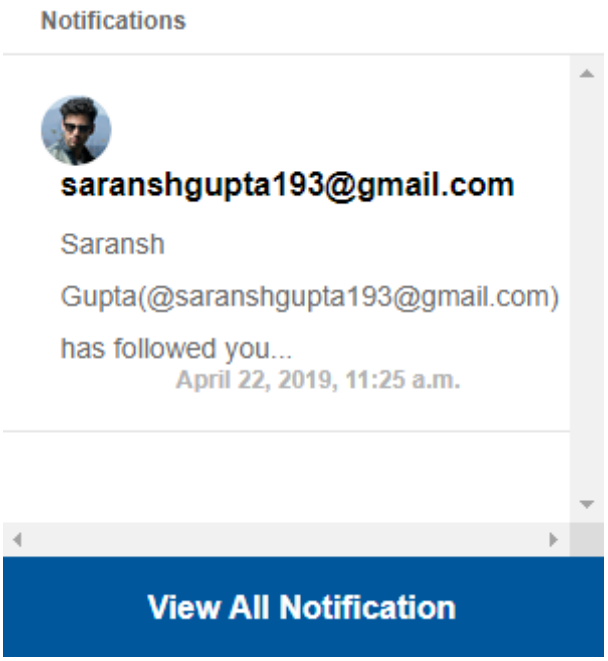


Fig 4.9 Following

4.5 Notification Module

This feature helps in notification alert to the registered people.

It pops up the notification of every following request, post and comments on post.



\

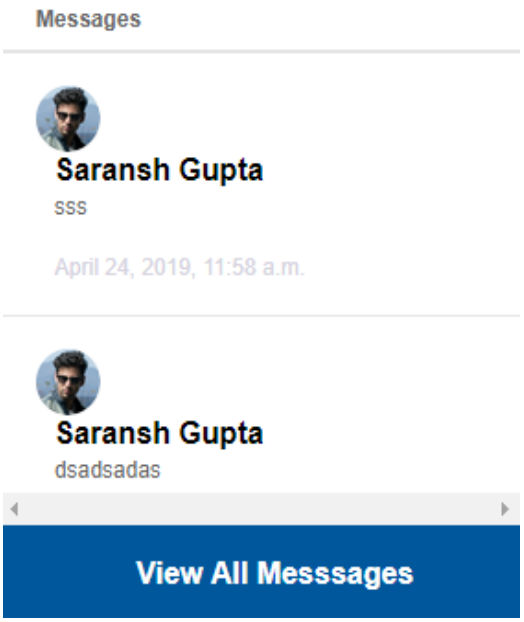


Fig 4.10 Notifications

Fig 4.11 Messages

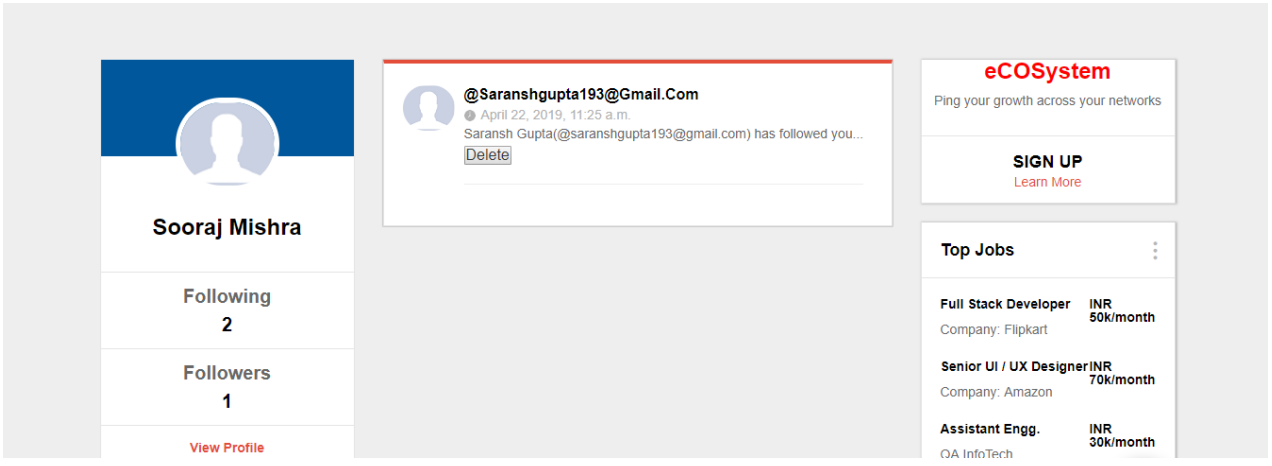


Fig 4.12 View All Notifications

4.6 Messaging/Chat Application

This feature helps the user in sending the text messages to other users.

He can send the text messages and images to his following and followers.

eCOSystem also provides the feature of group chat and group messaging.

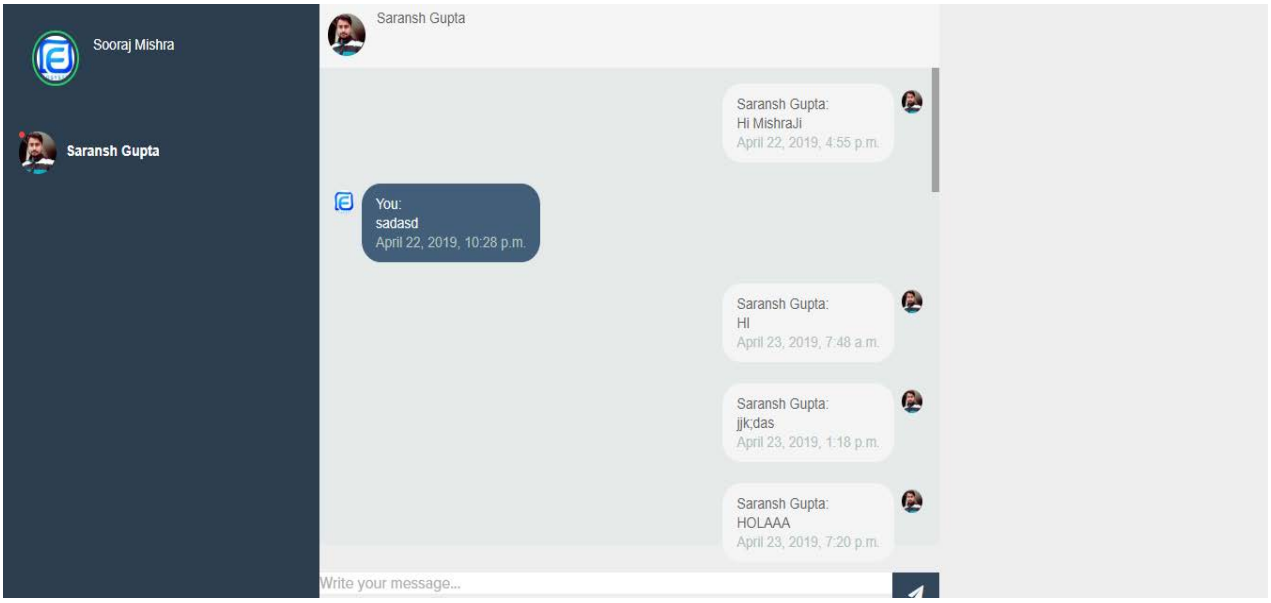


Fig 4.13 Chat Screen

4.7 Search (Advanced)

eCOSystem provides search feature to the users. Users can search the other users by name and by tags .user can also search the related post using the tags.

Advanced source searching can provide more comprehensive and accurate search results compared to simpler standard searches, which can be useful for the assessment and determination of topic notability.

This advance search technology makes easy interaction for users using particular keywords. Such as any user wants to search for GRE or GATE field then the user will get all the profiles related to these fields.

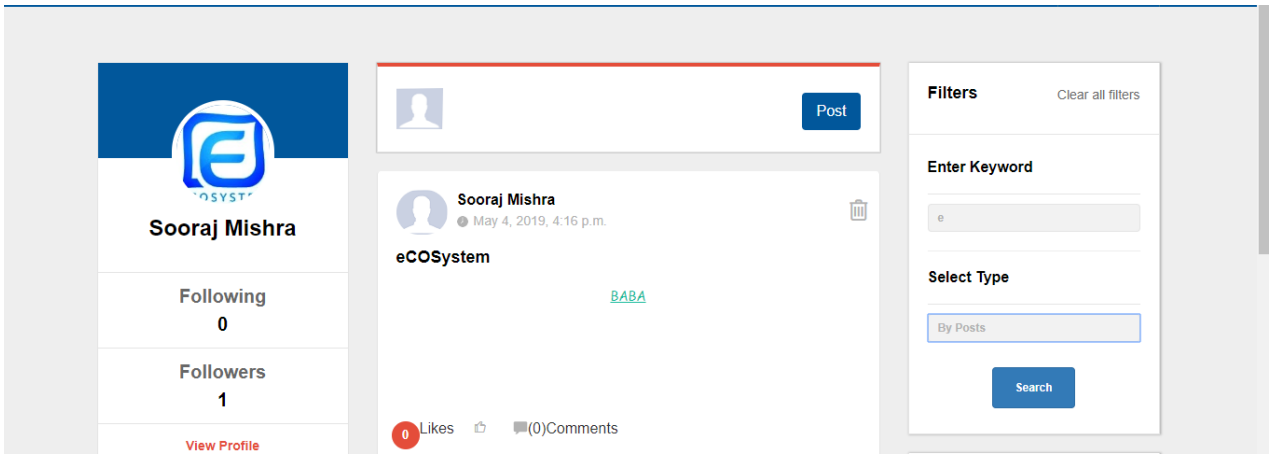


Fig 4.14 Advanced Search (Posts)

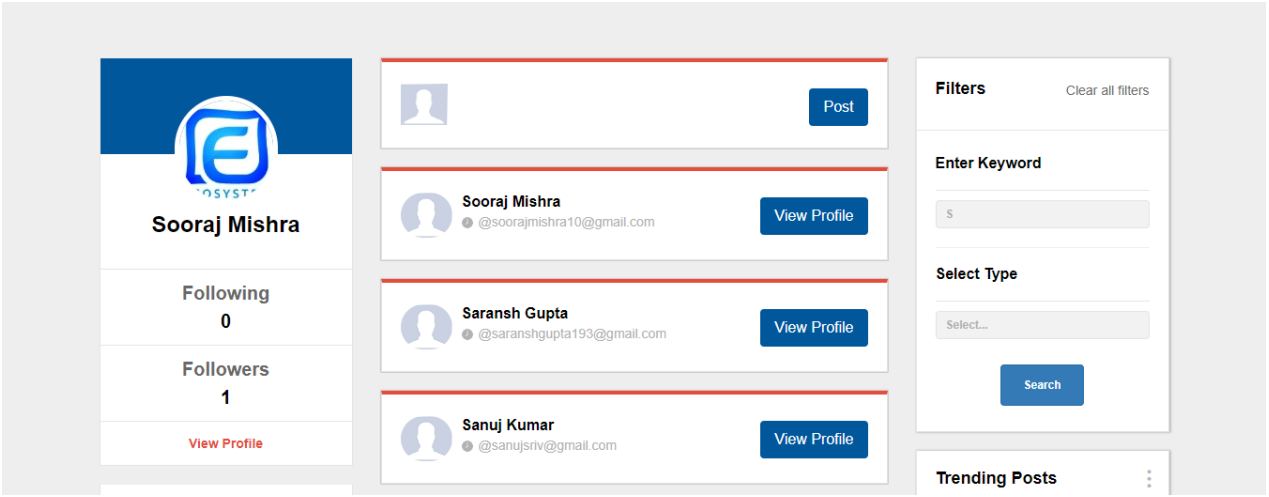


Fig 4.15 Advanced Search (Profiles)

4.8 Recommended Post Searches

In eCOSystem user can also search the recommended post according to his interest.

4.9 Profile Management

eCOSystem helps users to make profile. User can provide the details of his education, address, works, and his certification .users can view the profile details of the other users. He can also edit his profile any number of times.

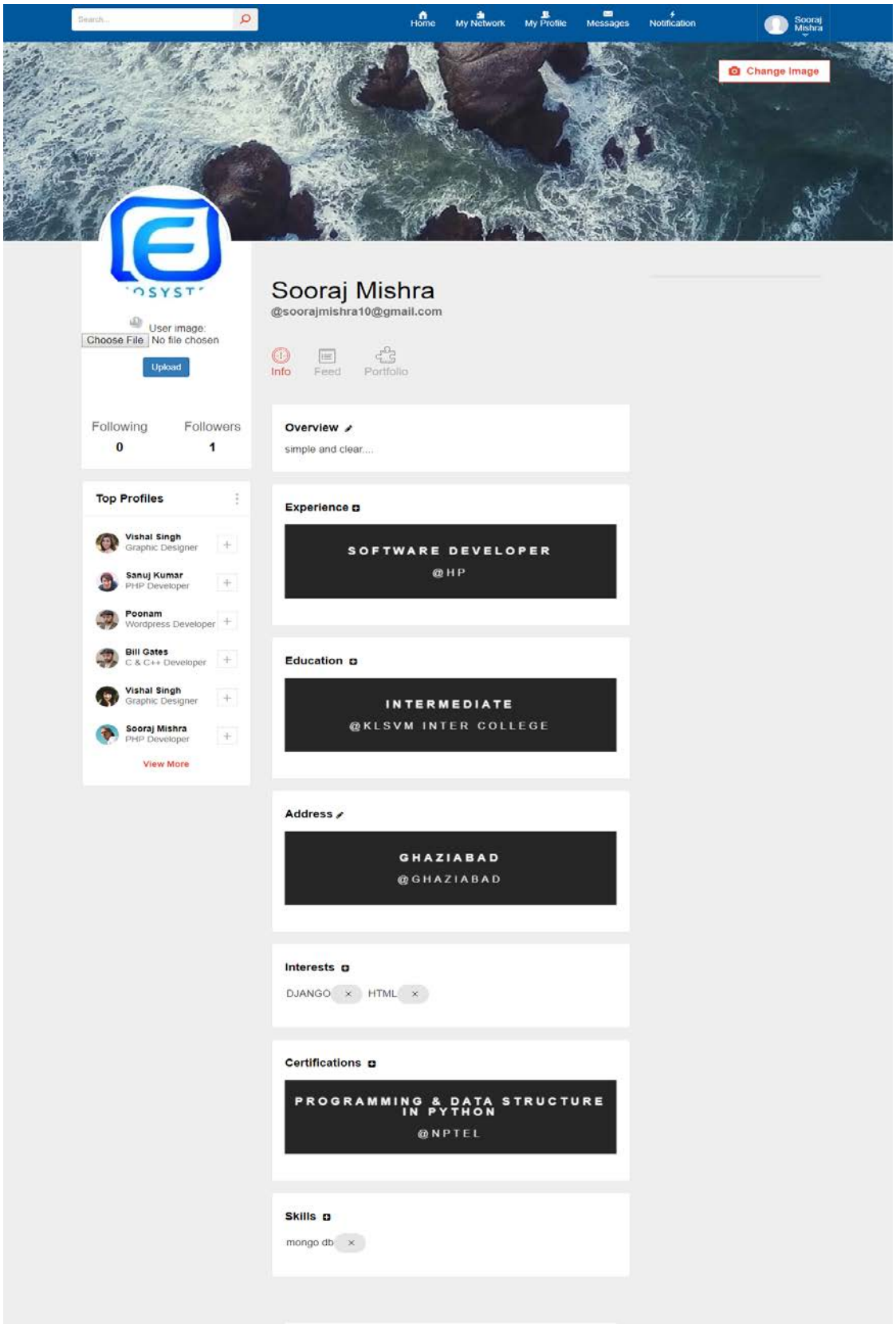


Fig 4.16 User Profile

4.10 Personalized Feeds

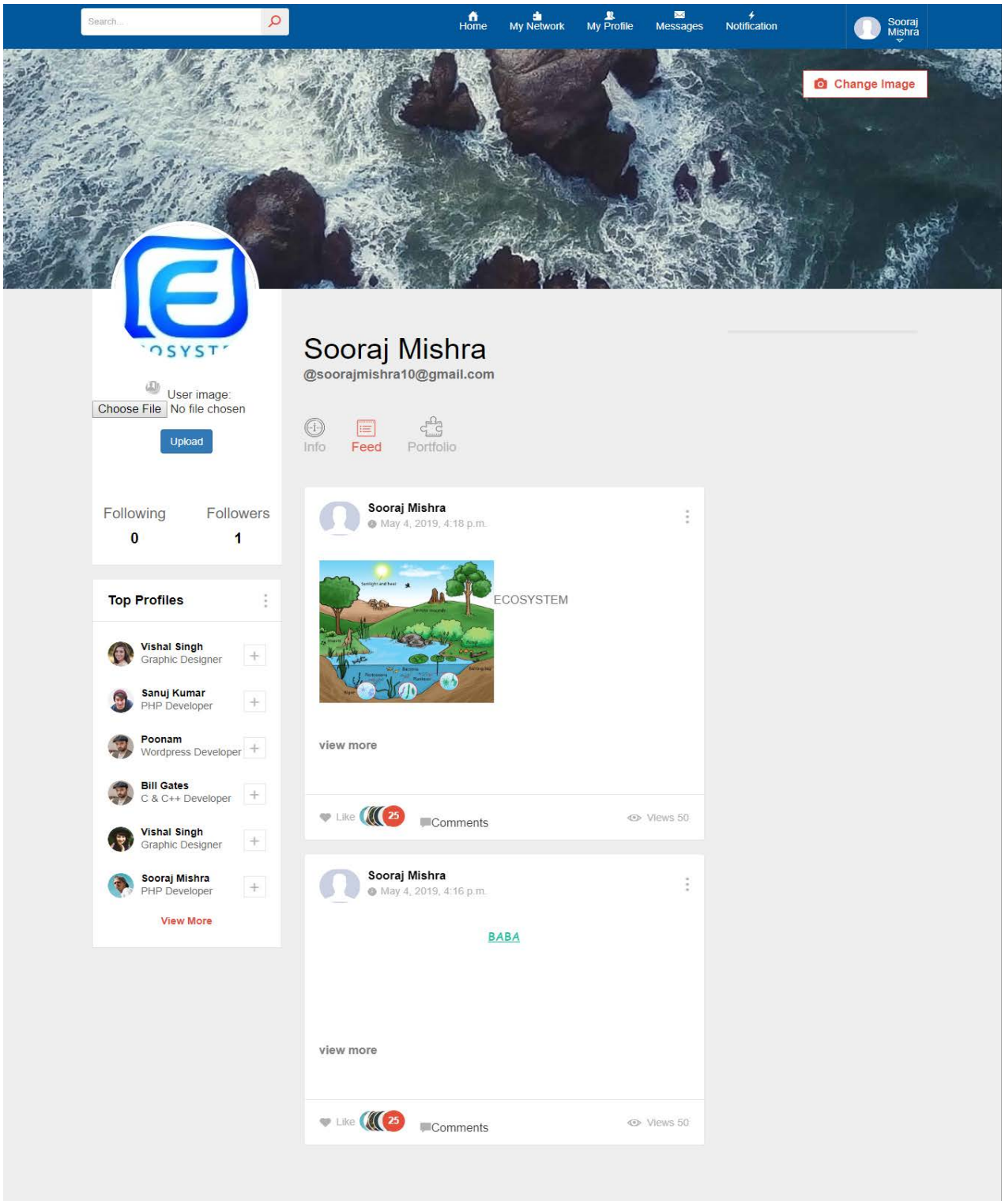
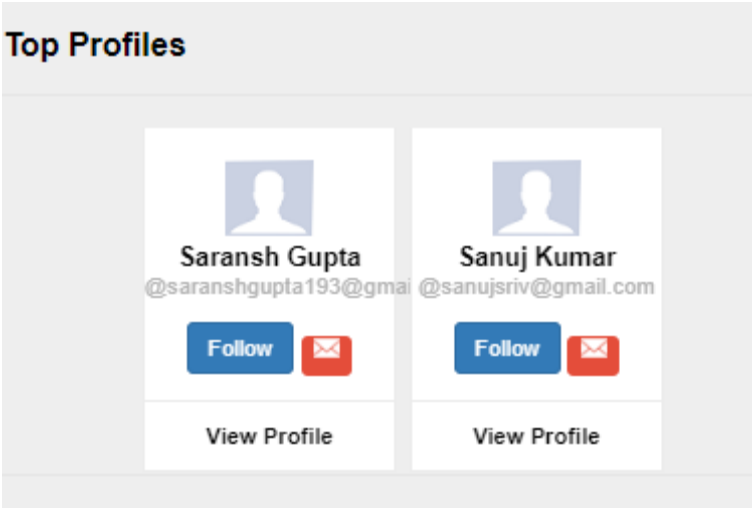


Fig 4.17 Personalized Feeds

4.11 Top Profiles

User specific profiles with most no. of followers is display as per your interest.



4.18 Top Profiles

4.12 Password Reset

Password Reset option is provided in case user forgets his/her password.

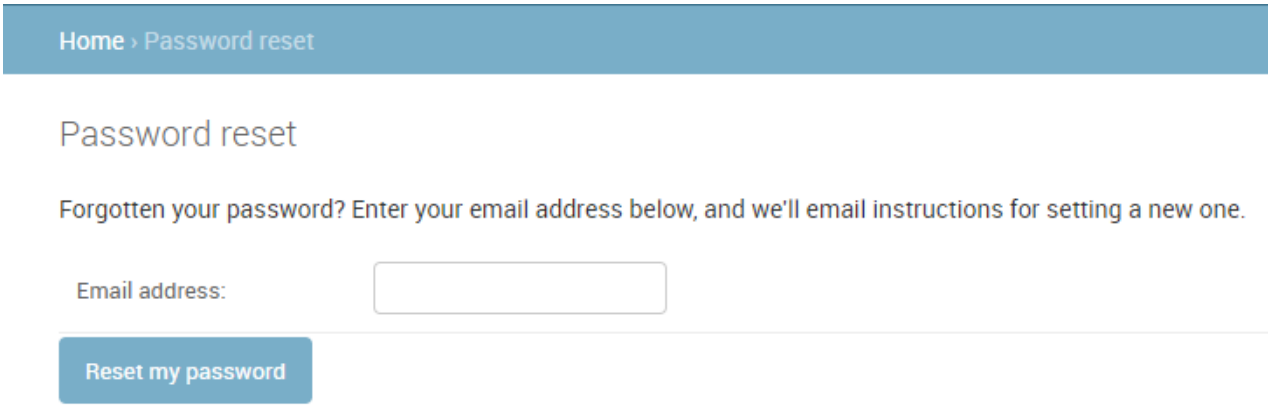


Fig 4.19 Password Reset (Authentication)

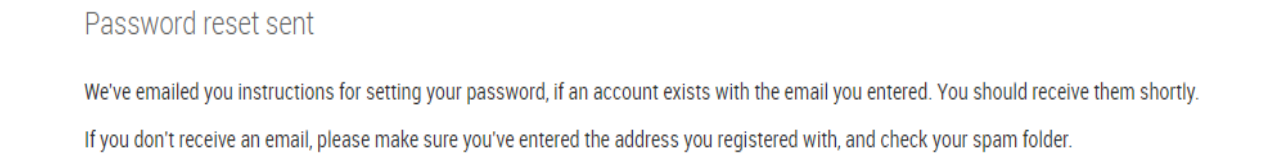


Fig 4.20 Password Reset (Confirmation)

4.13. Database

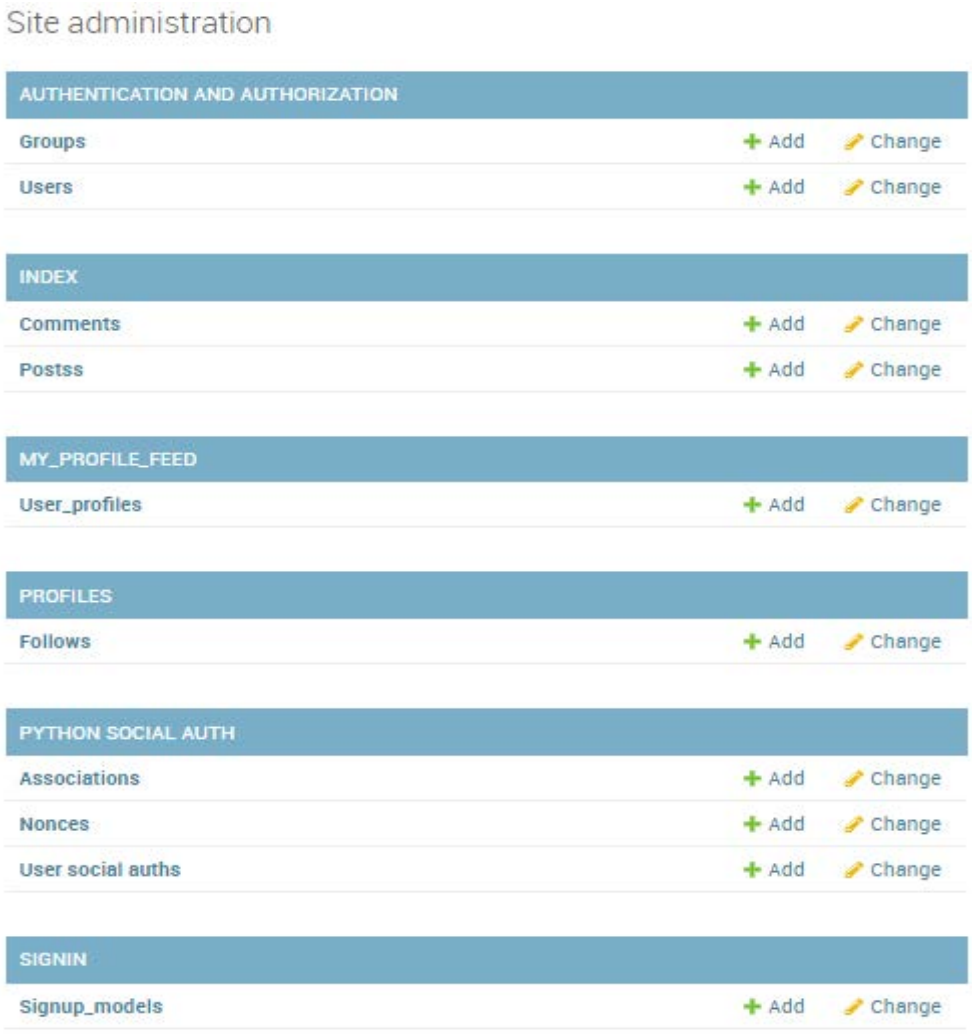


Fig 4.21 Django models

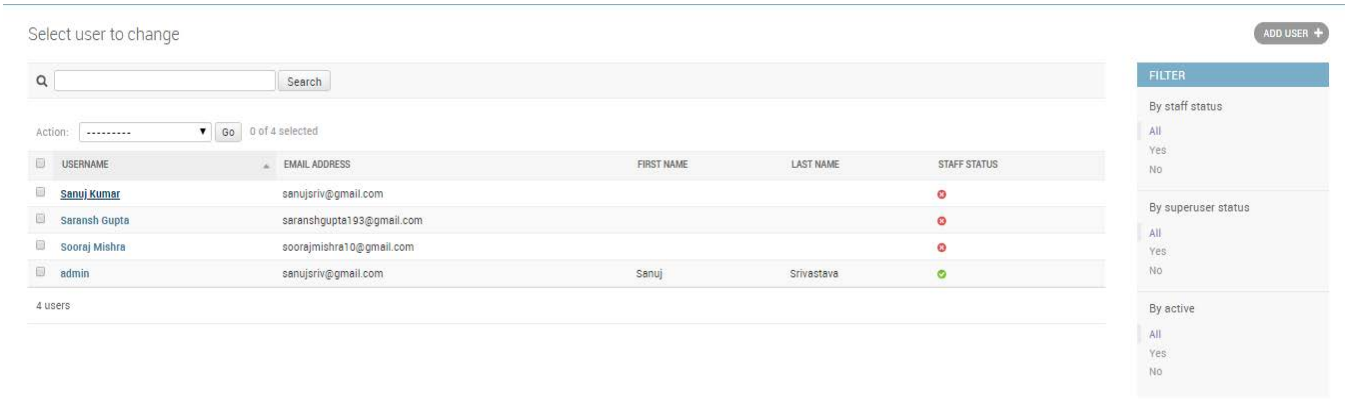


Fig 4.22 User Model

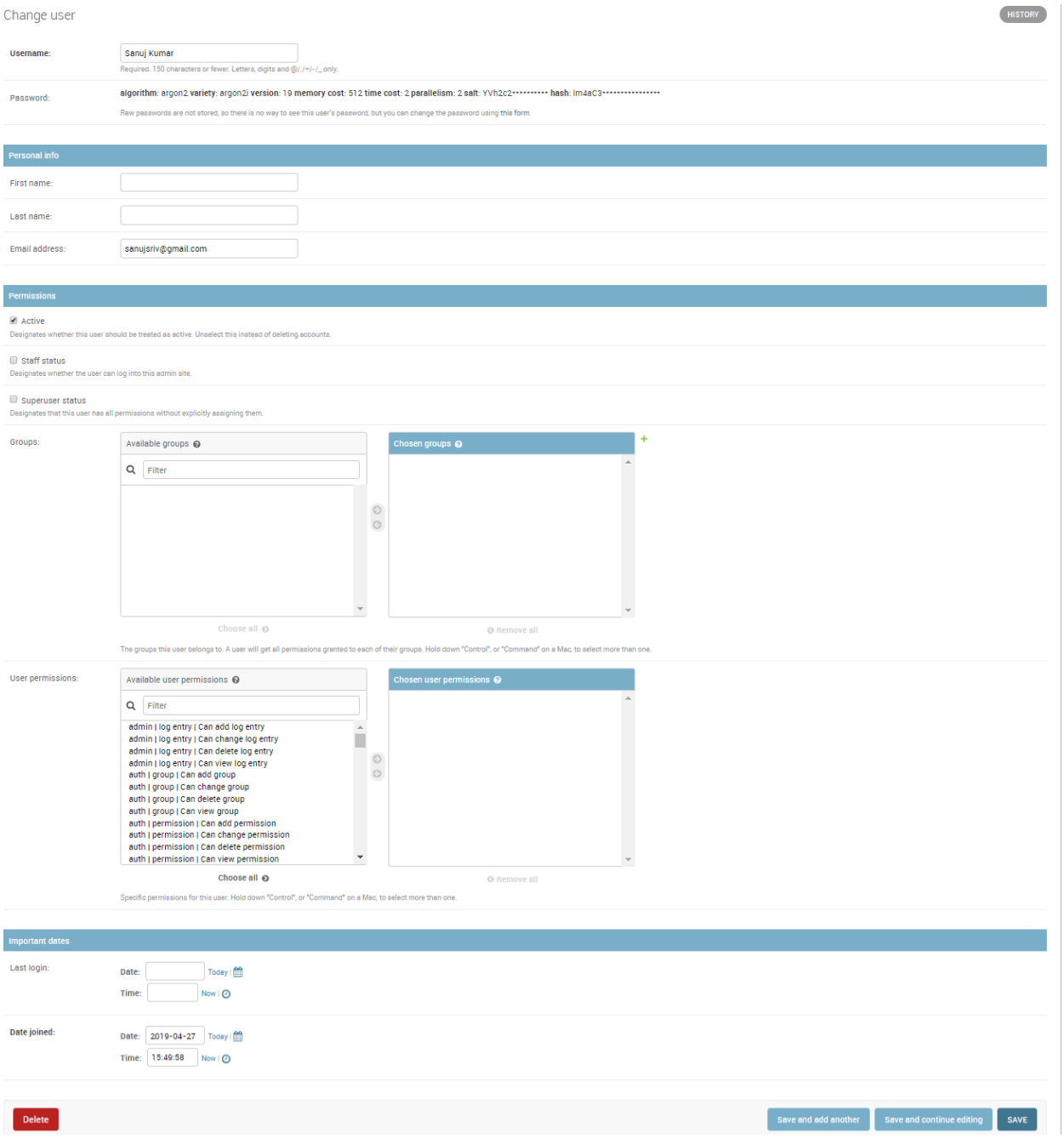


Fig 4.23 User Model Container

Fig 4.24 Posts Model

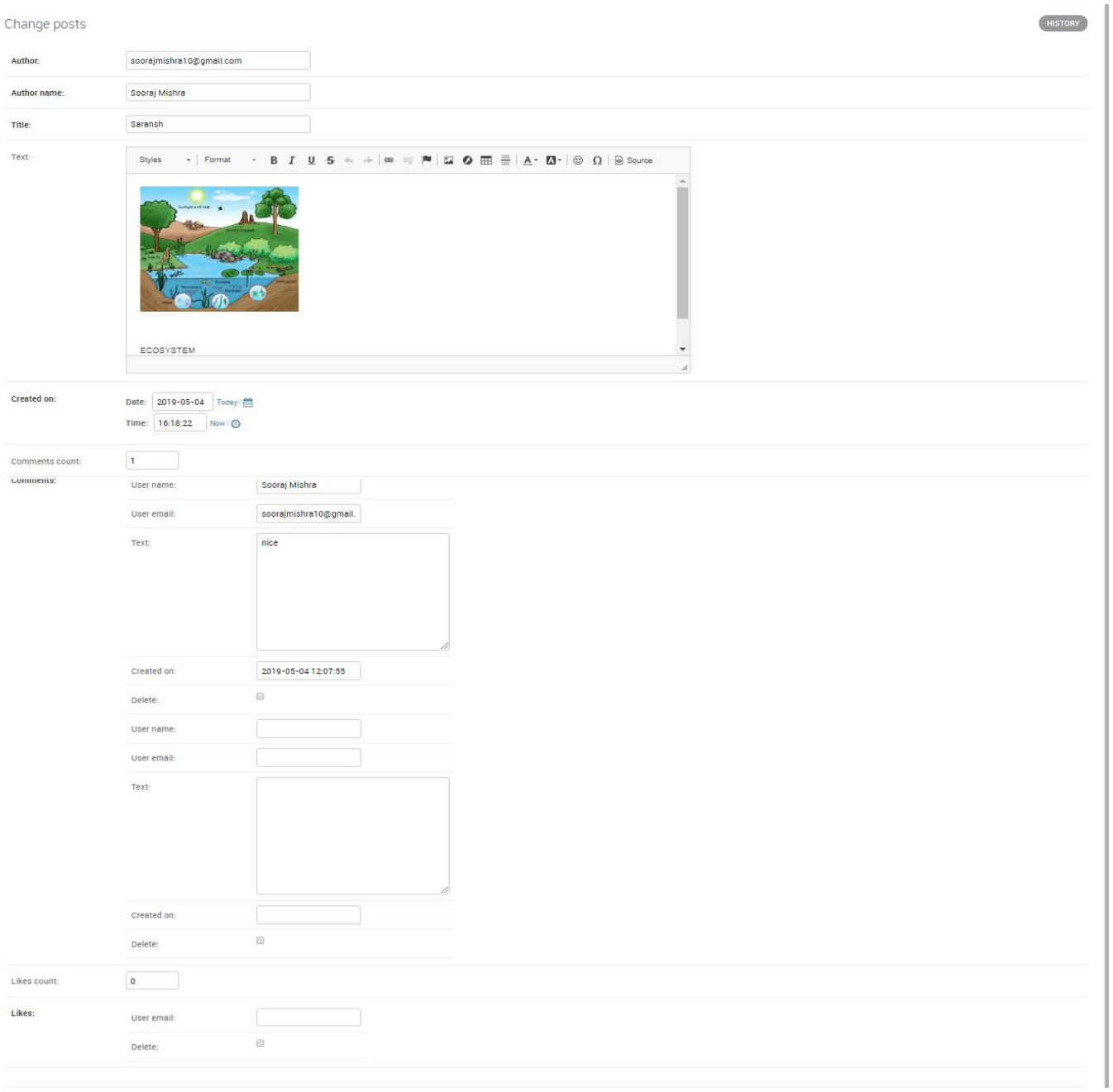


Fig 4.25 Posts Model Container

Fig 4.26 User Profile Model

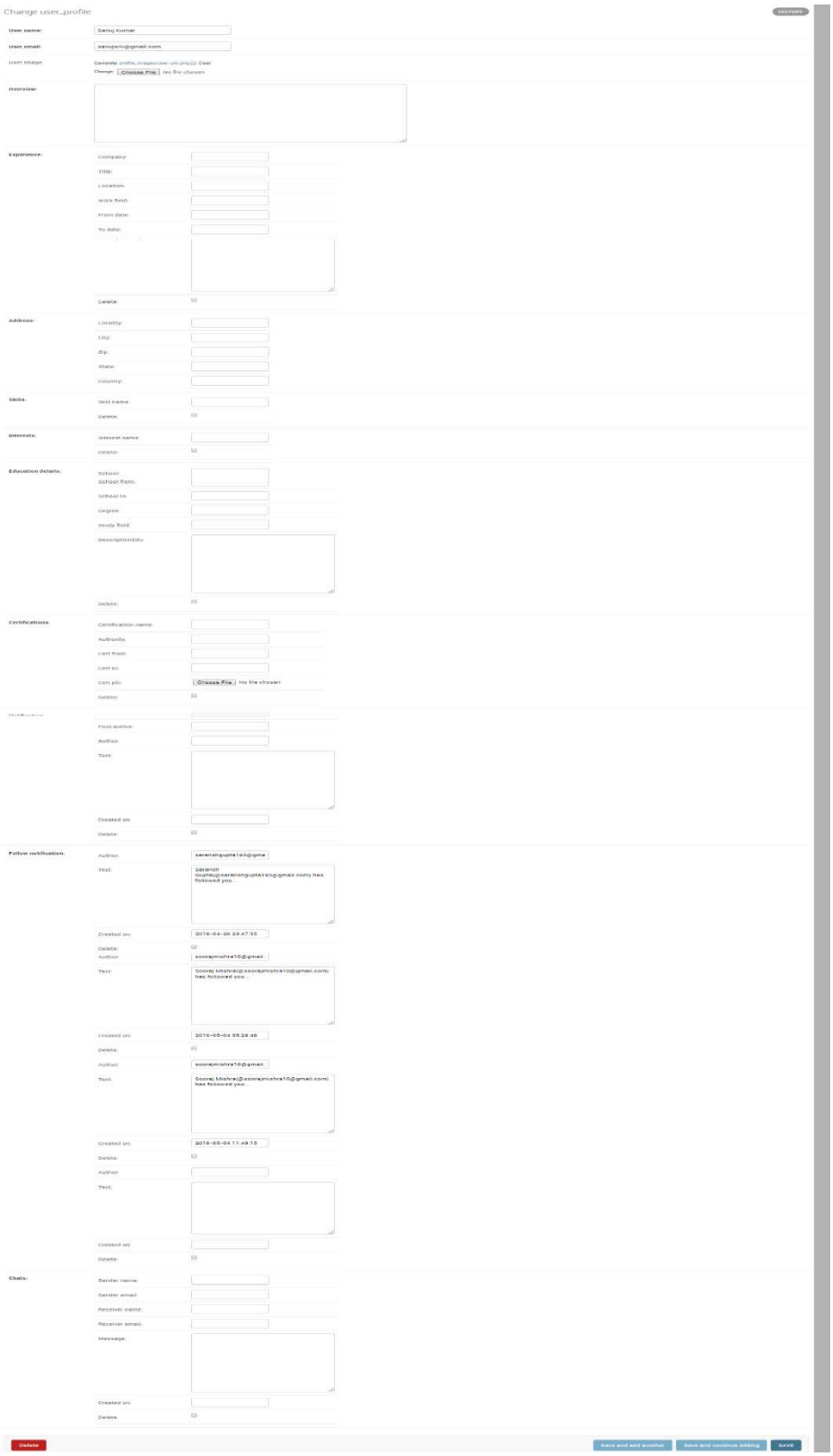


Fig 4.27 User Profile Model Container

Fig 4.28 Follow Model

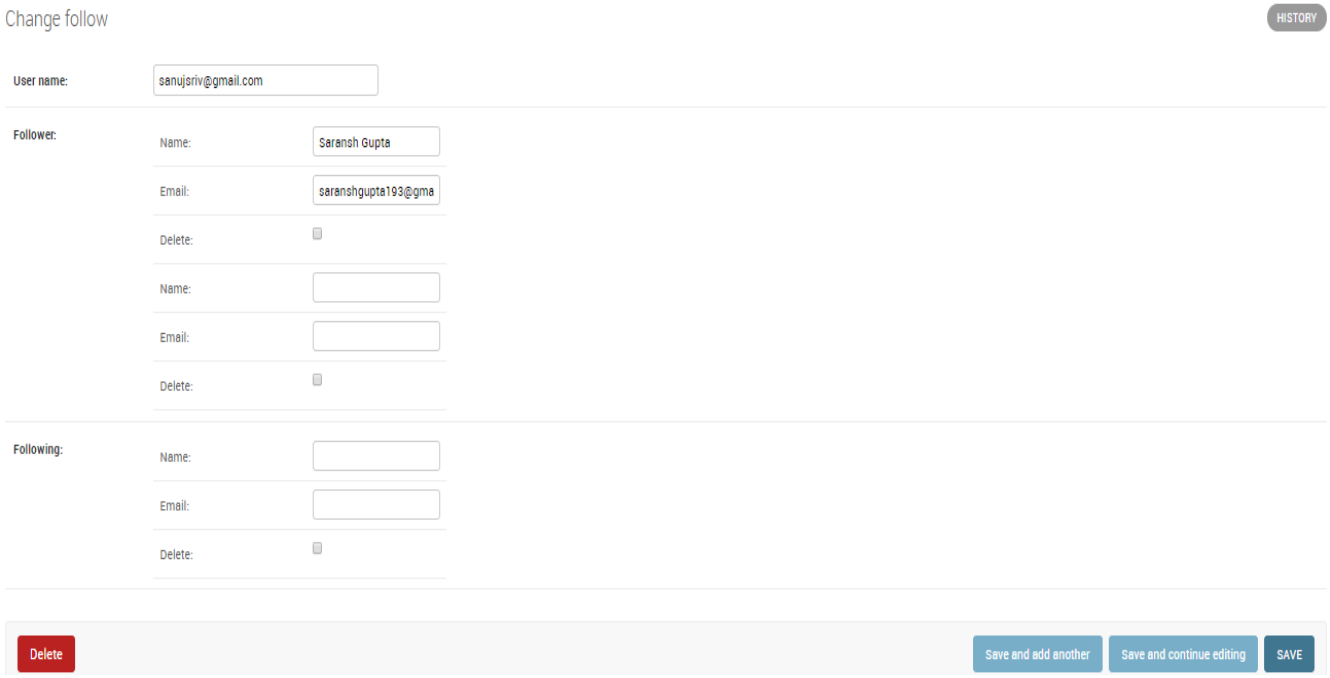


Fig 4.29 Follow Model Container

Fig 4.30 Signup Model

Fig 4.31 Signup Model Container

CHAPTER 5

METHODOLOGY

A software development methodology or system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system.

Feature Driven Development Methodology

Jeff De Luca and Peter Coad were both greatly involved in developing the Feature Driven Development methodology. Peter describes FDD as having just enough process to ensure scalability and repeatability while encouraging creativity and innovation

FDD is a model-driven short-iteration process that consists of five basic activities

- Develop overall model
- Build feature list
- Plan by feature
- Design by feature
- Build by feature

CHAPTER 6

IMPLEMENTATION

6.1 Django Modules in Project

- **Signin**
- **Index**
- **Profile**
- **My_profile_feed**

Signin:

This module is responsible for facilitating registration and sign in feature to the system. It saves the password of the registered user in hashed form [argon2] for making the system secure. Moreover, it also facilitates login via social authentication i.e. Google.

Workflow

The form seen in the frontend is actually applied from backend present in file **forms.py**. The styling of the form is done using **Django-crispy-form.** While returning the view from **views.py** the hashed password is compared with hashed value of entered password which on match returns **username** as __session__ in the dictionary of the render function as context.

Index:

This module is the main index page which facilitates basic user interaction such as:

- Post,
- General Search
- Advanced Search (By Profile)
- Advanced Search (By Post)
- Comment
- Like
- Followers and Followings

This module also handles displaying of searched profiles and posts

Workflow

The following view methods handles various functionalities of the index module:

PostListView: Returns view as list of post (recommended, user-specific, searched)

DislikeView: Returns view when user dislikes a post

LikeView: Returns view when user likes a post

CommentView: Returns view list of comments of a post

PostCommentView: Returns view when user comment on a post

Signout: Returns view when user logout

social_signout: Returns view when user sign out from social app

deletePostView: Returns view when user delete a post

allnotifications: Returns view collection of all notifications

deletenotifications: Returns view with deleted notifications

deleteCommentView: Returns view with deleted comments

Profile:

The profile module was developed for keeping record of all profiles. Personalized network for every user containing their followers and profiles they're following is handled by this module.

Workflow

The following view methods handles various functionalities of the profile module:

ProfileView: Returns view as list of searched profiles / recommended profiles / message and general notifications.

FollowingView: Returns view a list of profiles which you are following / notifications and messages of connected user's interactions

FollowersView: Returns view a list of profiles which are following you / notifications and messages of connected user's interactions

My_profile_feed:

This module is developed for personalized record maintenance. It facilitates the following features:

- User Profile/Dashboard
- Personalized feeds
- Upload Resume/Portfolio
- Change Profile Picture

Workflow

The following view methods handles various functionalities of the my_profile_feed module:

showchats: Returns view which shows chats to user after interaction.

sendChats: Returns view which shows chat interaction screen.

chatView: Returns view which shows chat menu screen.

my_profile_feed: Handles entire profile/dashboard , upload portfolio/profile picture, personalized feeds.

deleteDetailsView: Returns view which deletes the selected users' detail

customViewProfile: Returns view as entire screen of details of selected user

CHAPTER 7

TESTING

7.1. Login Module: Following test cases were considered to ensure the working of login module.

Table 1 - Login Module test case

Test case id	Test case	Expected output	Actual output	Test case result
1.1	Valid username valid password	Login success	Login success	**Pass**
1.2	Valid username invalid password	Login failure	Login failure	**Pass**
1.3	Invalid username valid password	Login failure	Login failure	**Pass**
1.4	Invalid username invalid password	Login failure	Login failure	**Pass**

7.2. Search module: Following test case were considered to ensure the working of search module.

Table 2 – Search Module test case

Test case id	Test case	Expected output	Actual output	Test case result
2.1	Searching valid profile/post	Redirect to the requested page	Redirect to the requested page	**Pass**
2.2	Searching invalid profile/page	"No such page found" message	"No such page found" message	**Pass**

7.3. Post Module: Following test cases were considered to ensure the working of post module.

Table 3 - Post Module test case

Test case id	Test case	Expected output	Actual output	Test case result
3.1	Posting content in **valid format**	Gets posted successfully	Gets posted successfully	**Pass**
3.2	Posting content in **invalid format**	Posts doesn't appear on feed	Posts doesn't appear on feed	**Pass**
3.3	Commenting on Valid Post	Post gets commented successfully	Post gets commented successfully	**Pass**
3.4	Commenting on Invalid Post	Redirected to index	Redirected to index	**Pass**
3.5	Handling of arbitrary 'null' posts in database	Null Posts gets deleted	Null Posts gets deleted	**Pass**

7.4. Notification Module: Following test cases were considered to ensure the working of notification module.

Table 4 - Notification Module test case

Test case id	Test case	Expected output	Actual output	Test case result
4.1	Notification of being followed	Notification received	Notification received	**Pass**
4.2	Notification on receiving new messages	Notification received	Notification received	**Pass**

4.3	Notification of post being liked by user	Notification received	Notification received	**Pass**
4.4	Notification on receiving comments for posts	Notification received	Notification received	**Pass**

7.5. Messaging Module: Following test cases were considered to ensure the working of messaging module.

Table 5 - Messaging Module test case

Test case id	**Test case**	**Expected output**	**Actual output**	**Test case result**
5.1	Messaging content in **valid format**	Message gets sent successfully	Message gets sent successfully	**Pass**
5.2	Receiving message from user	Message gets received successfully	Message gets received successfully	**Pass**
5.3	Sending message to user	Message gets sent successfully	Message gets sent successfully	**Pass**
5.4	Messaging allowed to only mutual connected parties	Allowed	Allowed	**Pass**
5.5	Enter key action using javascript	Successful	Successful	**Pass**

7.6. User Profile Module: Following test cases were considered to ensure the working of user profile module

Table 6 -User Profile Module test case

Test case id	Test case	Expected output	Actual output	Test case result
6.1	Filling user data in selected fields	Fields gets and stored in database filled selectively	Fields gets and stored in database filled selectively	**Pass**
6.2	Changing profile picture	Profile picture changes successfully	Profile picture changes successfully	**Pass**
6.3	Viewing personalized feeds	Personalized feeds	Personalized feeds	**Pass**
6.4	Uploading documents in portfolio	Personalized documents gets uploaded successfully	Personalized documents gets uploaded successfully	**Pass**
6.5	Viewing user-specific profiles	User-specific profiles	User-specific profiles	**Pass**

Further, testing was conducted to check for the entries/retrieval of data into/from the database. Passwords stored in the database table were in the encrypted MD5 format and thus password security is ensured. Post feed and messaging option were tested and found to work in the expected manner.

CHAPTER 8

DEPLOYMENT

AWS CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances running in your own facility, serverless AWS Lambda functions, or applications in an Amazon ECS service.

You can deploy a nearly unlimited variety of application content, such as an updated Lambda function, updated applications in an Amazon ECS service, code, web and configuration files, executables, packages, scripts, multimedia files, and so on. AWS CodeDeploy can deploy application content stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. You do not need to make changes to your existing code before you can use AWS CodeDeploy.

AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications, without many of the risks associated with error-prone manual deployments.

8.1 AWS CodeDeploy Components

Use the information in this guide to help you work with the following AWS CodeDeploy components:

- **Application**: A name that uniquely identifies the application you want to deploy. AWS CodeDeploy uses this name, which functions as a container, to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment
  .
- **Deployment group**: A set of individual instances, CodeDeploy Lambda deployment configuration settings, or an Amazon ECS service and network details. A Lambda deployment group specifies how to route traffic to a new version of a Lambda function. An Amazon ECS deployment group specifies the service created in Amazon ECS to deploy, a load balancer, and a listener to reroute production traffic to an updated containerized application. An EC2/On-premises deployment group contains individually tagged instances, Amazon EC2 instances in Amazon EC2 Auto Scaling groups, or both. All deployment groups can specify optional trigger, alarm, and rollback settings.

- **Deployment configuration**: A set of deployment rules and deployment success and failure conditions used by AWS CodeDeploy during a deployment.

- **Deployment:** The process and the components used when updating a Lambda function, a containerized application in an Amazon ECS service, or of installing content on one or more instances.

- **Application revisions**: For an AWS Lambda deployment, this is an AppSpec file that specifies the Lambda function to be updated and one or more functions to validate deployment lifecycle events. For an Amazon ECS deployment, this is an AppSpec file that specifies the Amazon ECS task definition, container, and port where production traffic is rerouted. For an EC2/On-premises deployment, this is an archive file that contains source content—source code, webpages, executable files, and deployment scripts—along with an AppSpec file. Revisions are stored in Amazon S3 buckets or GitHub repositories. For Amazon S3, a revision is uniquely identified by its Amazon S3 object key and its ETag, version, or both. For GitHub, a revision is uniquely identified by its commit ID.

8.2. CodeDeploy

CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

You can deploy a nearly unlimited variety of application content, including:

- code
- serverless AWS Lambda functions
- web and configuration files
- executables
- packages
- scripts
- multimedia files

CodeDeploy can deploy application content that runs on a server and is stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. CodeDeploy can also deploy a serverless

Lambda function. You do not need to make changes to your existing code before you can use CodeDeploy.

CodeDeploy makes it easier for you to:

- Rapidly release new features.
- Update AWS Lambda function versions.
- Avoid downtime during application deployment.
- Handle the complexity of updating your applications, without many of the risks associated with error-prone manual deployments.

The service scales with your infrastructure so you can easily deploy to one instance or thousands.

CodeDeploy works with various systems for configuration management, source control continuous integration, continuous delivery, and continuous deployment. For more information, see Product Integrations.

8.3. How to integrate with Django WSGI

Django's primary deployment platform is WSGI, the Python standard for web servers and applications.

Django's startproject management command sets up a simple default WSGI configuration for you, which you can tweak as needed for your project, and direct any WSGI-compliant application server to use.

The key concept of deploying with WSGI is the application callable which the application server uses to communicate with your code. It's commonly provided as an object named application in a Python module accessible to the server.

The startproject command creates a file <project_name>/wsgi.py that contains such an application callable.

It's used both by Django's development server and in production WSGI deployments.

WSGI servers obtain the path to the application callable from their configuration. Django's built-in server, namely the runserver command, reads it from the WSGI_APPLICATION setting. By default, it's set to <project_name>.wsgi.application, which points to the application callable in <project_name>/wsgi.py.

Configuring the settings module

When the WSGI server loads your application, Django needs to import the settings module — that's where your entire application is defined.

Django uses the DJANGO_SETTINGS_MODULE environment variable to locate the appropriate settings module. It must contain the dotted path to the settings module. You can use a different value for development and production; it all depends on how you organize your settings.

If this variable isn't set, the default wsgi.py sets it to mysite.settings, where mysite is the name of your project. That's how runserver discovers the default settings file by default.

Since environment variables are process-wide, this doesn't work when you run multiple Django sites in the same process. This happens with mod_wsgi.

To avoid this problem, use mod_wsgi's daemon mode with each site in its own daemon process, or override the value from the environment by enforcing os.environ["DJANGO_SETTINGS_MODULE"] = "mysite.settings" in your wsgi.py.

To apply WSGI middleware you can simply wrap the application object. For instance you could add these lines at the bottom of wsgi.py:

```
from helloworld.wsgi import HelloWorldApplication

application = HelloWorldApplication(application)
```

You could also replace the Django WSGI application with a custom WSGI application that later delegates to the Django WSGI application, if you want to combine a Django application with a WSGI application of another framework.

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

Acquiring industrial knowledge is not an option for a fresher but having some idea related to the current scenario and requirements of an industry/corporate is important. Only a candidate with some experience can list out the exact details for all the other candidates. This system is built with the idea for make such details reachable to all those require so they can prepare themselves.

Everyone wants to leave behind legacy for everyone else to acknowledge so as to share their experiences, the challenges faced and their decisions so as leave a mark behind for others to know and learn. This is a vital feature of our system.

Knowing what to learn at the right time and what you should learn for furthering your career in a definite stream is very important. Through this system we can be guided by many individuals with similar interest who have already been through this stage and will provide directions in a better way so that we can make the right choice.

9.2 FUTURE SCOPE

- Implementation of such a platform in any institute would be much more beneficial for its culture.
- The system is itself a knowledge repository which is improved and updated by regularly.
- The system can used a social networking platform.
- Encourages open source development and connectivity with live projects.
- Promotes awareness and knowledge.

Some real-world application of our project are as following

- **Social Networking**

The system can work as a platform for social interaction and data sharing in addition to its features of providing relevant information and making career choices.

- **Counselling and Guidance**

The system features a “guide” mechanism which creates a network of individuals with similar interest therefore promoting collective growth and experience.

- **Open Source Development & Live Project**

Candidates working in the same field with similar skill set could have an hands-on experience on live projects. Therefore the system provides a much better experience to individuals by introducing them to open source development

- **Follow and Learn**

The system features a “follow” which attaches a network of individuals with similar interest thereby instigating the learning process of inexperienced candidates.

REFERENCES & BIBLIOGRAPHY

Jaiswal, Sanjeev; Kumar, Ratan (22 June 2015), Learning Django Web Development (1st ed.), Packt, p. 405, ISBN 1783984406

Ravindrun, Arun (31 March 2015), Django Design Patterns and Best Practices (1st ed.), Packt, p. 180, ISBN 1783986646

Osborn, Tracy (May 2015), Hello Web App (1st ed.), Tracy Osborn, p. 142, ISBN 0986365912

Bendoraitis, Aidas (October 2014), Web Development with Django Cookbook (1st ed.), Packt, p. 294, ISBN 178328689X

Baumgartner, Peter; Malet, Yann (2015), High Performance Django (1st ed.), Lincoln Loop, p. 184, ISBN 1508748128

Elman, Julia; Lavin, Mark (2014), Lightweight Django (1st ed.), O'Reilly Media, p. 246, ISBN 149194594X

Percival, Harry (2014), Test-Driven Development with Python (1st ed.), O'Reilly Media, p. 480, ISBN 1449364829

Django Essentials, Samuel Dauzon

Daniel Greenfeld and Audrey Roy, Two Scoops of Django

Zed A. Shaw, Learn Python the Hard Way

https://simpleisbetterthancomplex.com/tutorial/2016/10/24/how-to-add-social-login-to-django.html

https://djongo.readthedocs.io/
https://python-social-auth-docs.readthedocs.io/en/latest/configuration/settings.html

https://searchcode.com/codesearch/view/94440820/

https://stackoverflow.com/questions/13018147/authalreadyassociated-exception-in-django-social-auth

https://stackoverflow.com/questions/14529815/logout-with-django-social-auth
https://phppot.com/php/comments-system-using-php-and-ajax/
https://pypi.org/project/django-threadedcomments/
https://github.com/v1k45/django-notify-x
https://stackoverflow.com/questions/5135556/dynamic-file-path-in-django
https://samulinatri.com/blog/django-ckeditor-codesnippet-hightlightjs-youtube/
https://ckeditor.com/cke4/builder
https://docs.recombee.com/tutorial.html
https://docs.recombee.com/api.html
https://simpleisbetterthancomplex.com/tutorial/2016/08/01/how-to-upload-files-with-django.html
https://django-mongodb-engine.readthedocs.io/en/latest/topics/gridfs.html
https://stackoverflow.com/questions/38065944/uploading-file-as-id-using-models-ind-django
https://coderwall.com/p/mvsoyg/django-dumpdata-and-loaddata
https://themeforest.net/item/workwise-the-freelancer-and-social-networking-html-template/21647879
https://pypi.org/project/bcrypt/
https://pypi.org/project/argon2/
https://pypi.org/project/pytz/
https://pypi.org/project/recombee-api-client/
https://pypi.org/project/PyYAML/
https://pypi.org/project/Pillow/
https://pypi.org/project/text2html/
https://pypi.org/project/textblob/
https://www.udemy.com/python-and-django-full-stack-web-developer-bootcamp/
https://colorlib.com/wp/bootstrap-social-network-templates/
https://simpleisbetterthancomplex.com/tutorial/2018/11/28/advanced-form-rendering-with-django-crispy-forms.html
https://docs.djangoproject.com/en/1.7/topics/db/queries/#field-lookups-intro
https://django-mongodb-engine.readthedocs.io/en/latest/tutorial.html
https://django-mongodb-engine.readthedocs.io/en/latest/topics/setup.html
https://django-mongodb-engine.readthedocs.io/en/latest/topics/index.html

https://code.djangoproject.com/wiki/NoSqlSupport
https://simpleisbetterthancomplex.com/tutorial/2018/11/28/advanced-form-rendering-with-django-crispy-forms.html
https://djongo.readthedocs.io/docs/array-model-field/
https://nesdis.github.io/djongo/array-model-field/#example
https://docs.djangoproject.com/en/2.1/ref/models/querysets/#create
https://pypi.org/project/Faker/