**A**

**PROJECT REPORT**

**ON**

**NEWS BY TAG**

**(http://newsbytag.herokuapp.com)**

B.Tech (CE) Sem-VI

In fulfillment of all requirements for

**Bachelor of Technology**

**In**

**Computer Engineering**

**SEM VI**

**In the Subject of**

**System Design Practice**

**Rajat N Movaliya (CE-66) (ID:15CEUOS109)**

Under the Guidance of

**Prof. Jigar M Pandya**

****

**DEPARTMENT OF COMPUTER ENGINEERING**

**FACULTY OF TECHNOLOGY,**

**DHARMSINH DESAI UNIVERSITY**

COLLEGE ROAD, NADIAD- 387001

**DHARMSINH DESAI UNIVERSITY NADIAD-387001,**

**GUJARAT**

****

**CERTIFICATE**

This is to certify that the project carried out in the subject of

System Design Practice entitled “NEWS BY TAG” and recorded in

this report is a work of

|  |  |  |
| --- | --- | --- |
| Rajat Movaliya | Roll No: CE-66 | ID:15CEUOS109 |

in Department of Computer Engineering, semester VI. He was

involved in Project developing during the academic year 2017 -2018.

Prof. Jigar M Pandya Dr. C.K.Bhensdadia

(Project Guide), CE Dept., Head, CE Dept.,

Faculty of Technology, Faculty of Technology,

Dharmsinh Desai University, Dharmsinh Desai University,

Nadiad Nadiad

**CONTENTS**

**Abstract** **i**

[**1. Introduction**](#page6) **1**

[**1.1 Project details: Brief Introduction**](#page6) **2**

[**1.2 Technology Used**](#page6) **4**

**2.** [**Software Requirement Specifications**](#page7) **6**

[**2.1 Purpose**](#page7) **7**

[**2.2 Scope**](#page7) **7**

[**2.3 O**](#page7)**verall Description 7**

[**2.3.1 P**](#page8)**roduct Perspective 7**

[**2.3.2 P**](#page8)**roduct Functions 7**

[**2.3.3 User Classes and Characteristics**](#page8) **8**

[**2.3.4 Operating Requirements**](#page8) **8**

[**2.3.5 D**](#page9)**esign and Implementation Constraints 8**

[**2.3.6**](#page9) **Assumptions and Dependencies 9**

[**2.4**](#page9) **External Interface Requirements 9**

[**2.4.1**](#page9) **User Interfaces 9**

[**2.4.2**](#page9) **Hardware Interfaces 10**

[**2.4.3**](#page9) **Software Interfaces 13**

[**2.4.4**](#page9) **Communication Interfaces 14**

[**2.5**](#page9) **Funtional Requirements 14**

[**3. Design**](#page10) **15**

[**3.1**](#page10) **Activity Network Diagram 16**

[**3.2**](#page10) **Gantt Chart 17**

[**3.3 E**](#page10)**R Diagram 18**

[**3.4 Data Dictionary**](#page10) **20**

[**4. Implementation**](#page13) **21**

[**4.1 Implementation Environment**](#page13) **22**

[**4.2 Modules Description**](#page13) **23**

[**5. Testing**](#page14) **24**

[**5.1 Testing Plan**](#page14) **25**

[**5.2 Testing**](#page14) **Strategy 25**

[**5.3 Testing**](#page14) **Methods 26**

[**5.4 Test**](#page14) **Cases 28**

[**6.**](#page14) **Screenshots 29**

[**7.**](#page14) **Software Version Deployment Steps 39**

[**8.**](#page14) **Limitations and Future Enhancements 41**

[**9.**](#page14) **Conclusion 43**

[**10.**](#page14) **Bibliography 45**

**Abstract**

In the era of digitization, everybody has access to online news via various websites or applications, which shows an overview of all types of breaking-news, that’s fine with regular use, but in this type, user irritate with many irrelevant new so many users everytime searches their favorite topics on the internet which is waste of time.

So the solution to this limitation is one can tell their favorite topics and system will store that topics in that user’s account and give only news relevant to that respective topic. Also that topics will be updated frequently, so when the user comes next time, directly from user’s account user can select any previously entered topic and see new breaking-news.

This is the aim of this project and by this, it will save the time of the user and increase knowledge in the direction user wants.

**CHAPTER: 1**

**INTRODUCTION**

1. **Introduction**

**1.1** **Project Details: Brief Introduction**

News By Tag is a web application. The main aim of developing this system is to ease out the process of reading relevant news by giving only news related to topics user wants.

The system will store all favorite topics of the user and update it, everytime new news comes respective to that topic, so the user does not have to go everytime everywhere in internet to just find news of his/her favorite topic.

**1.2** **Technology Used**

**Front End :**

* **HTML**
* **CSS**
* **JavaScript (jQuery (v3.3.1))**
* **Bootstrap (v3.3.7)**

As the system is a web application, the need for HTML, CSS, and JavaScript is understood.

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers

Bootstrap is an open source toolkit for developing front-end of a web app with HTML, CSS, and JavaScript.

**Back End :**

* **Python (v3.6.4) ( Flask Framework (v0.12.2))**
* **SqlAlchemy (v1.2) (ORM Module of Python)**
* **News API (v2)**

The project is built using Flask framework of python with SqlAlchemy as ORM module (Object Relational Mapping) to handle back-end side of database manipulation.

NewsAPI is for fetching new news of particular topics with HTTP requests and getting a response of news in JSON (JavaScript Object Notation) format.

**Database : PostgreSQL (v10.3)**

PostgreSQL is a powerful, open source object-relational database system. Classified as a semi-structured database program, PostgreSQL (latest version) supports both SQL and NoSQL operation. In the project, only SQL part is used.

**API Testing Tool: Postman**

To make API development simple while testing and processing request and response.

**Version Control Tool :**

* **Git (VCS) (v2.17)**
* **GitKraken (GUI Tool for Git)**

Git is a [free and open source](https://git-scm.com/about/free-and-open-source) distributed VCS (Version Control System) designed to track modification in project files with versions.

GitKraken is GUI (Graphical User Interface) tool to handle all simple git operation with clicks instead of commands in the terminal in a very fast and effective manner.

**Deployment Platform: Heroku (PaaS)**

Heroku is a cloud platform (Platform as a Service) that lets the user build, deliver, monitor and scale apps with the support of CI ( Continuous Integration )

**Diagram Environment: ERD-Plus**

ER-Diagram of this project is made in online web app ERD-Plus.

**Operating System: Ubuntu (17.10)**

Ubuntu is an open-source operating system for computers. It is a Linux Distribution based on the Debian architecture and is used as the development environment of this project.

**CHAPTER: 2**

**SOFTWARE REQUIREMENTS**

**SPECIFICATIONS ( SRS )**

**2.** **Software Requirement Specifications**

**2.1 Purpose**

This is an SRS document refers to News By Tag Release 2017 version 1.0.0. It describes the functionality and specification of how the System will help the user save time by providing only news they want in an efficient way. It also specifies how it will interact with the end users.

**2.2 Scope**

This system will provide the functionality of adding a new tag to account and system with showing trending tag based on how many user stores that tags.

This system will not provide a general overview of all kind of live breaking news as it is provided by most of the other web application like the-hindu, the-times-of-india etc.



**2.3 Overall Description**

**2.3.1 Product Perspective**

This software is developed as a part of coursework the subject “System Design Practice”. The software aims to provide only news which user wants with ease. The web- app‘s the main perspective is towards saving the time of the user and providing a user-flexible system of online news reading.

**2.3.2 Product Functions**

The product offers a breaking news related to tags entered by the user in their account, once the tag is added to the system it will be updated every day.

The user can add new tag directly by writing or by searching tag (entered by another user) or also from trending tag section.

The system won’t allow the user to add a tag which is vulgar, currently support for three languages English, Gujarati, Hindi in profanity filter. Admin can add any tag to block list if he/she find it vulgar.

**2.3.3** **User Classes and Characteristics**

1. News Reader:-

The system is developed for people of all genre and mainly for those who love to read news and blog post of specific topics without searching it from various sources.

1. Admin:-

Admin of this web app can, delete any user, if find it fake and can delete any tag or add to block list if find it vulgar.

**2.3.4 Operating Environment**

Because the system is developed based on RESTful API concept, where host app will provide all back-end task which can be fired by HTTP requests, the system can support all kind of web API consumer interface like web client (chrome, firefox, opera etc.) or Android app or ios app etc.

Currently, there is support for web client (various browsers). Because of the front-end development of web client is responsive, it can adjust UI (User Interface) to any screen size.

**2.3.5 Design and Implementation Constraints**

The main constraint is that there is not every time news exists for rare tags, so when user enter some rare tags he/she won’t get any news for a long time.

Some vulgar tags may not be identified by the system, so admin has to manually add it to block list category, which is a time-consuming process.

**2.3.6 Assumptions and Dependencies**

The system may not update new news if the appropriate network is not available because it completely depends on fetching new news from API request.

The system requires the user to be online because web client can’t store that much of data as offline.

**2.4** **External Interface Requirements**

**2.4.1 User Interfaces**

The application can be accessed through any browser interface. The software will be fairly compatible with Microsoft Internet Explorer Version 6 and above or other modern web browsers as well.

**2.4.2 Hardware Interfaces**

**Server Side**:

Processor: Pentium 3.0 GHz or higher.

RAM: 2GB or higher.

Hard-Disk: 100GB or more.

**Client Side:**

Processor: Pentium 3.0 GHz or higher.

RAM: 2GB or higher.

Hard-Disk: 100GB or more.

**2.4.3 Software Interfaces**

**Client Side:**

HTML supported Web Browsers, Windows 7,8, 8.1, 10 ,MAC OS, Linux (All Flavors).

**Server Side:**

Linux, PostgreSql Client.

**2.4.4 Communication Interfaces**

The “News By Tag” shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

**2.5** **Functional Requirements**

**R1**: System provides a feature to add new tags to the account.

**Input**: Tag Details

**Output**: Stored tag details with the status

**Processing**: Given tag name will be added to the database and will be sent for a queue of fetching news related to that tag and user will be redirected to that tag page.

**R2**: System provides a feature prevent from adding new profanity word as the tag name.

**Input**: Tag Details

**Output**: Status of profanity

**Processing**: It matches the list of profanity word and if matched then according to status will be given.

**R3**: System will provide facility to show trending tags.

**Input**: All tags details with user link

**Output**: Ordered list of trending tags

**Processing**: It will order the tags based on a number of users added that tag to their account.

**R4**: System will rank articles based on relevancy to tag name in a best possible way.

**Input**: Tag with corresponding articles

**Output**: Modified link order of article with the tag

**Processing**: Based on indexing algorithm it will sort the articles with relevancy level to that tag and modify that order.

**R5**: System provides a feature prevent from adding new profanity word as the tag name

**Input**: Tag Details

**Output**: Status of profanity

**Processing**: It matches the list of profanity word and if matched then according to status will be given.

**R6**: System provides the basic functionality of sign up, sign in and change the password.

**Input**: Credentials, action

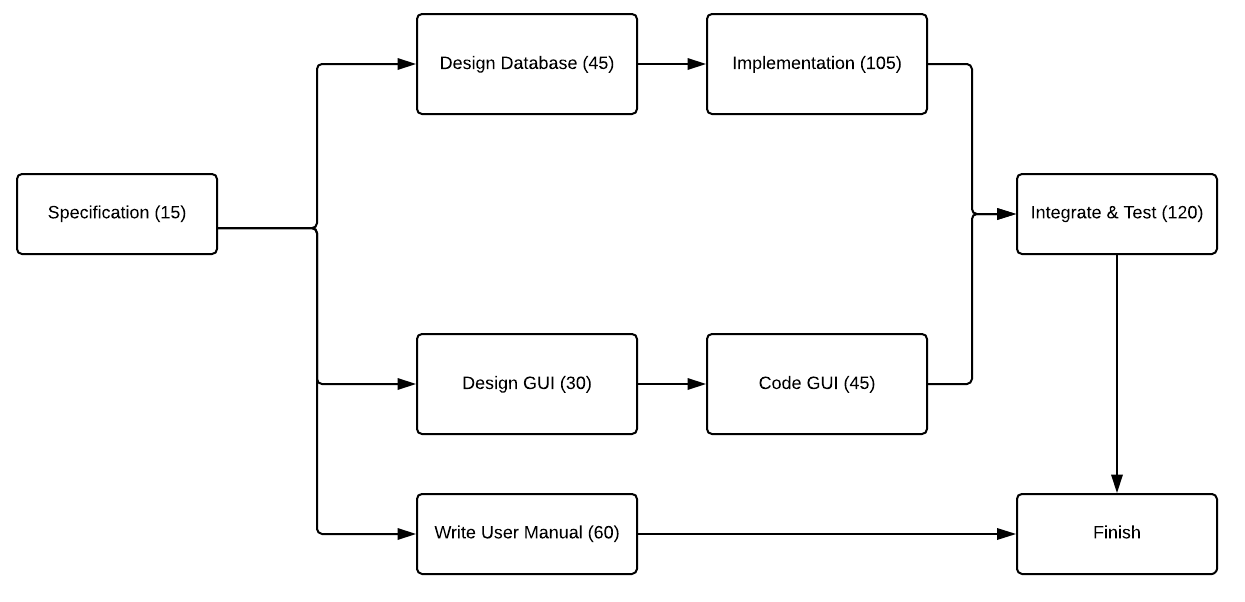
**Output**: Status

**Processing**: It will take log in, sign in or changing password related action based on the input of credential and action.

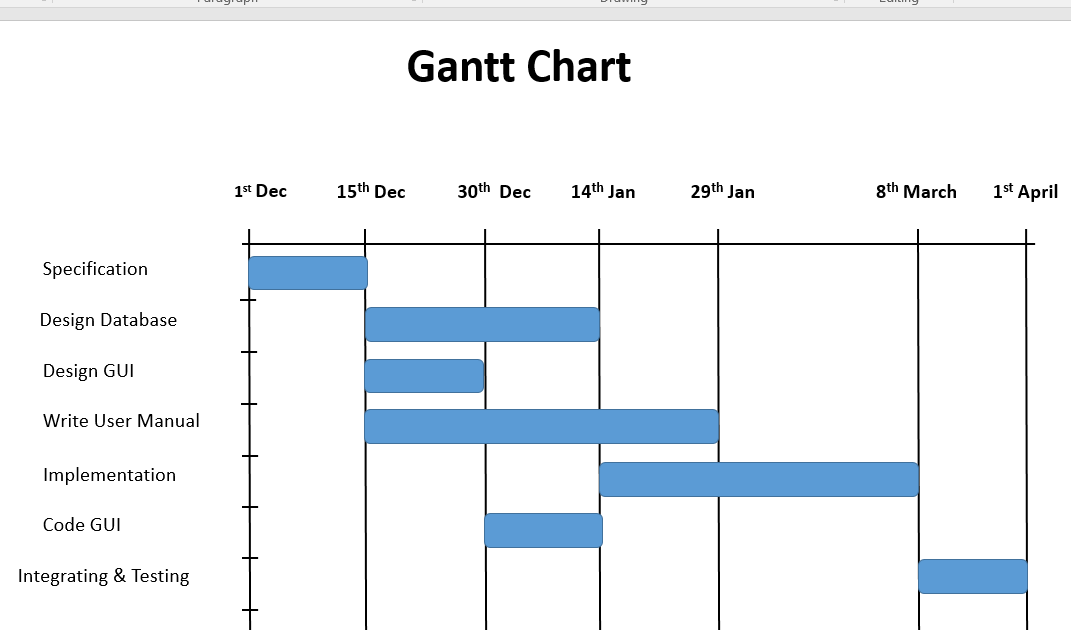
**CHAPTER : 3**

**DESIGN**

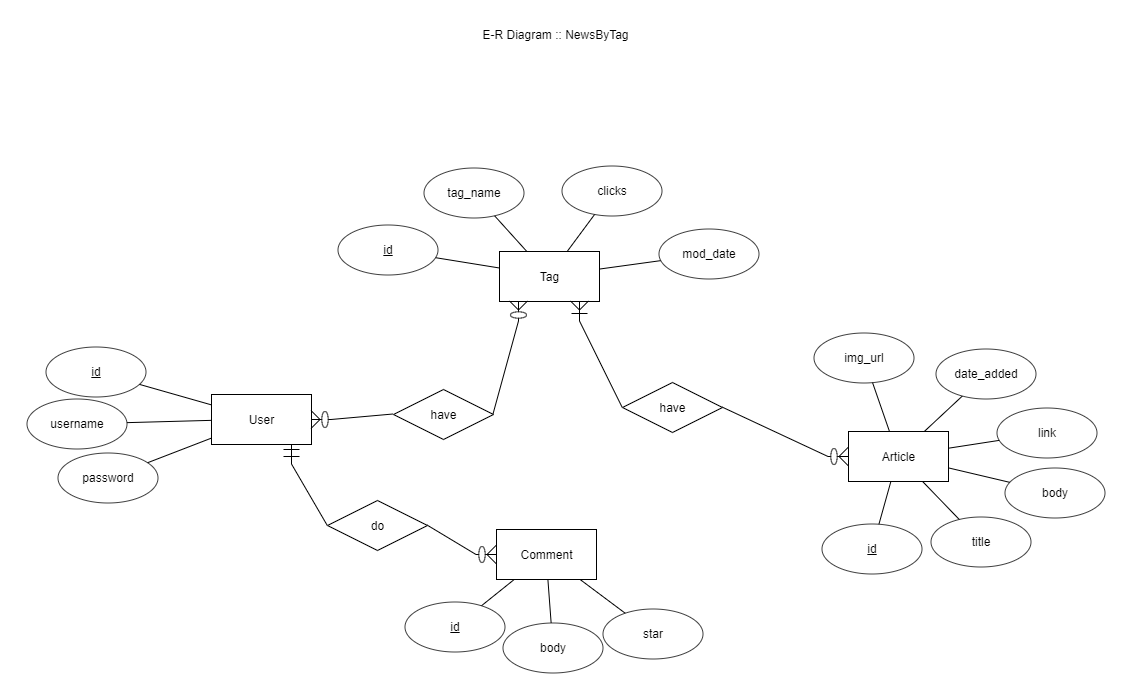
**3.1 Activity Network Diagram**

****

**3.2 Gantt Chart**

****

**3.3 E-R Diagram**

****

**3.4 Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User** | | | | |
| **Sr.no** | **Field\_name** | **Data\_type** | **Width** | **Constraint** |
| 1 | Id | Int |  | PRIMARY KEY  UNIQUE |
| 2 | Username | Varchar | 20 | NOT NULL |
| 3 | Password | Varchar | 20 | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tag** | | | | |
| **Sr.no** | **Field\_name** | **Data\_type** | **Width** | **Constraint** |
| 1 | Id | Int |  | PRIMARY KEY  UNIQUE |
| 2 | Tag\_name | Varchar | 20 | NOT NULL |
| 3 | Clicks | Int |  | NOT NULL |
| 4 | Mod\_date | DateTime |  | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User\_Tag** | | | | |
| **Sr.no** | **Field\_name** | **Data\_type** | **Width** | **Constraint** |
| 1 | User\_id | Int |  | NOT NULL  FOREIGN KEY |
| 2 | Tag\_id | Int |  | NOT NULL  FOREIGN KEY |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Article** | | | | |
| **Sr.no** | **Field\_name** | **Data\_type** | **Width** | **Constraint** |
| 1 | Id | Int |  | PRIMARY KEY  UNIQUE |
| 2 | Title | Varchar | 200 | NOT NULL |
| 3 | Body | Varchar | 1000 | NOT NULL |
| 4 | Img\_url | Varchar | 1000 | NOT NULL |
| 5 | Link | Varchar | 1000 | NOT NULL |
| 6 | Date\_added | DateTime |  | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tag\_Article** | | | | |
| **Sr.no** | **Field\_name** | **Data\_type** | **Width** | **Constraint** |
| 1 | Article\_id | Int |  | NOT NULL  FOREIGN KEY |
| 2 | Tag\_id | Int |  | NOT NULL  FOREIGN KEY |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Comment** | | | | |
| **Sr.no** | **Field\_name** | **Data\_type** | **Width** | **Constraint** |
| 1 | Id | Int |  | PRIMARY KEY  UNIQUE |
| 2 | Body | Varchar | 200 | NOT NULL |
| 3 | Star | Int |  | NOT NULL |
| 4 | User\_id | Int |  | NOT NULL  FOREIGN KEY |

**CHAPTER: 4**

**IMPLEMENTATION**

**4. Implementation**

**4.1** **Implementation Environment**

* PostgreSQL Client in Ubuntu
* Python3 environment in Ubuntu
* Git Client in Ubuntu
* Postman and GitKraken in Ubuntu

**4.2 Modules Description**

* **Sign-Up / Log - In Module**

It is used to store information of users who are new to the system and accessing the system for the first time and also authenticating the users before they can explore various auctions listed in the system.

**Input**: User’s information or credentials

**Output**: Stored or Verified successfully

**Processing**: Check user’s credentials in the database while logging in or store them in the database while registering a new user.

* **Tag Module**

This module comes into action when a user wants to access or store tag related information.

**Input**: Username, Tagname, Action

**Output**: Status of Action in JSON

**Processing**: It will add, delete or modify data about tag as given in action parameter.

* **News Module**

It will fetch news from the database and return back all result articles of given tag.

**Input**: Tagname

**Output**: Status of Result and News in JSON

**Processing**: It will return all the articles related to given tag name and return back to JSON format.

* **Updater Module**

It will update new entered tags as well as renew already entered tags with new news by fetching news from API request and make the link between the tag and that article.

**Input**: Tagname

**Output**: Status of Result

**Processing**: It will link tag with the corresponding article fetched from API request and store it to database

* **Admin Control Module**

This module is used by the administrator of the system in order to maintain control of the present tag, user, and comment. By this admin can delete any tag and user and also add a tag to block list category to avoid next time addition of that tag to the system it admin finds it vulgar.

**CHAPTER: 5**

**TESTING**

1. **Testing**

**5.1 Testing Plan**

The testing is a technique that is going to be used in the project is black box testing, the expected inputs to the system are applied and only the outputs are checked.

**5.2 Testing Strategy**

The development process repeats this testing subprocess a number of the lines for the following phases.

* Unit Testing
* Integration Testing

Unit Testing tests a unit of code after coding of that unit is completed. Integration Testing tests whether the previous programs that make up a system, interface with each other as desired. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification.

Testing is carried out in such a hierarchical manner to that each component is correct and the assembly/combination of the component is correct. Merely testing a whole system at the end would most likely throw up errors in a component that would be very costly to trace and fix. We have performed both Unit Testing and System Testing to detect and fix errors.

**5.3 Testing Methods**

We have performed Black-box testing for the testing purpose. A brief description is given below:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system, and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

**5.4 Test Cases**

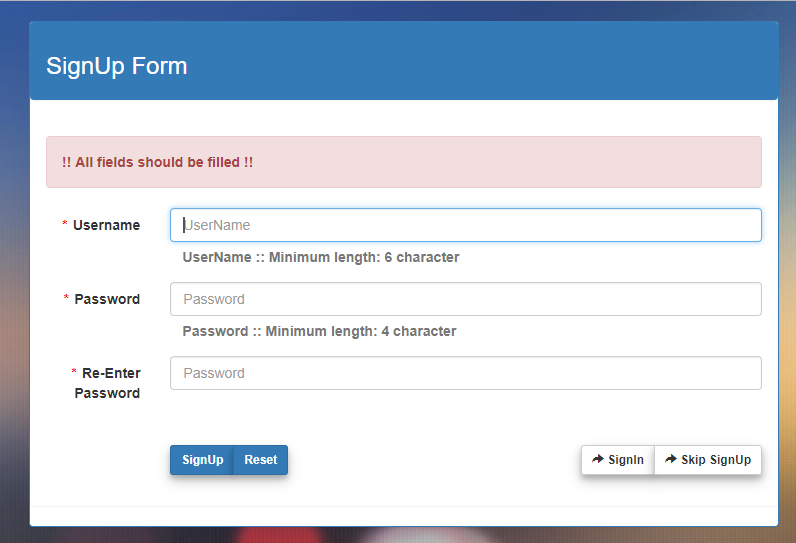
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test**  **Case ID** | **Test**  **Scenario** | **Test**  **Steps** | **Test**  **Data** | **Expected**  **Results** | **Actual**  **Results** |
| T01 | Sign Up  with proper data | 1.Provide right  Information on the signup page | User  Information | Stored  Successfully | Success |
| T02 | Sign Up  with empty as well as wrong data | 1. In signup page, enter wrong details | User  information | Display a respective error message | Error messages  displayed |
| T03 | Log In  with proper data | 1.Enter proper credential | Login Credentials | Display Homepage as logged in | Success |
| T04 | Log In  with improper data | 1. In the login page, enter wrong details | Login Credentials | Display a respective error message | Error messages  displayed |
| T05 | Adding proper New Tag | 1. Enter proper tag name with keeping the limit in character | Tag Name | Adding a tag to the database and fetching new news | Success |
| T06 | Adding Profanity word | 1.Add Improper profanity word | Profanity word as a tag name | Showing error message accordingly | Error messages  displayed |
| T07 | Tag Limit Per Account | 1.Add tags to limit in one account  2.Enter one more tag | Tag Limit per Account | Showing error message accordingly | Error messages  displayed |
| T08 | Changing password with the wrong old password | 1. Enter the wrong old password | Password matching capability | Showing according to msg | Error messages  displayed |
| T09 | Changing password with the right old password | 1.Enter right old password | Password matching capability | The message of the changed password on the homepage | The message displayed on the homepage |
| T10 | Deleting a tag from the account | 1.Click Delete the icon from my tag section | Deleting tag | The tag should not appear that in my tag section | Success |
| T11 | Adding the tag to the block list | 1.Open Admin Panel  2.Click add icon near any tag | Adding a tag to block list | Tag name should be shown in the blacklist and should not be added next time in the system | Success |

**CHAPTER: 6**

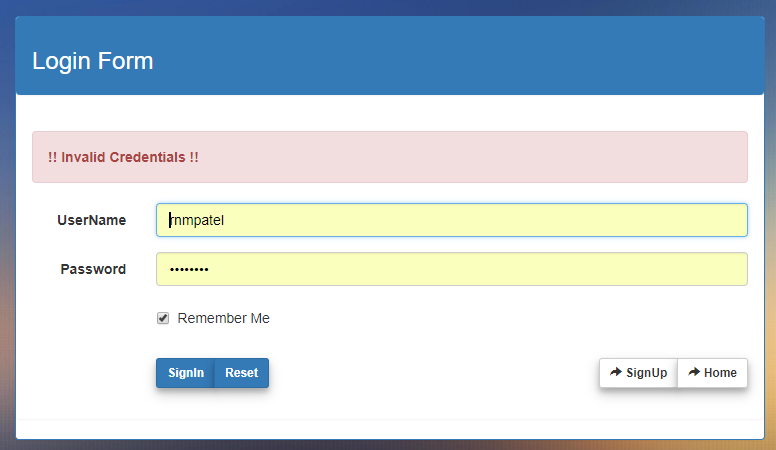
**SCREENSHOTS**

**6. Screen-shots**

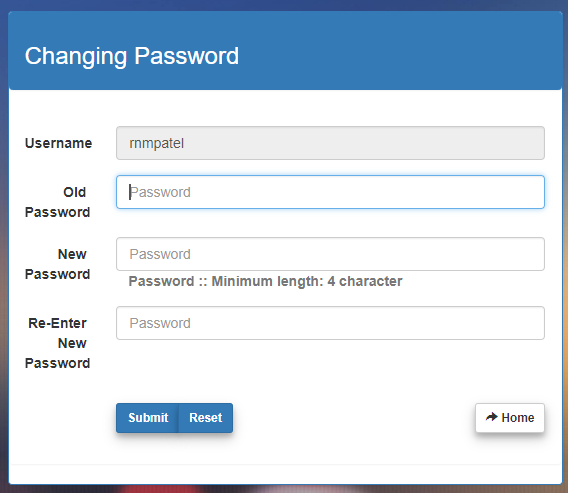
6.1 SignUp, log in, Change Password and FeedBack



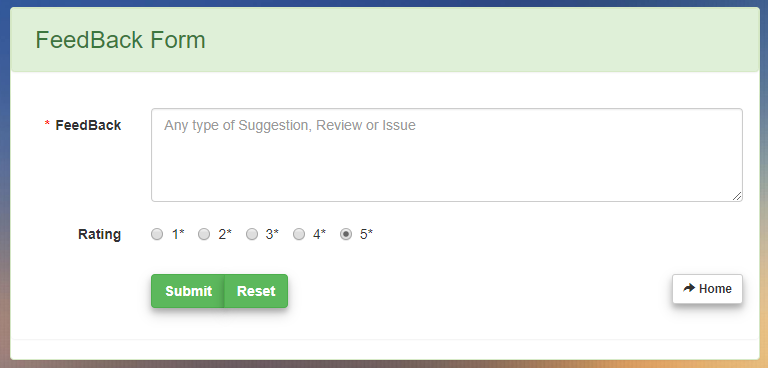
**Sign-Up Page**



**Log-In Page**

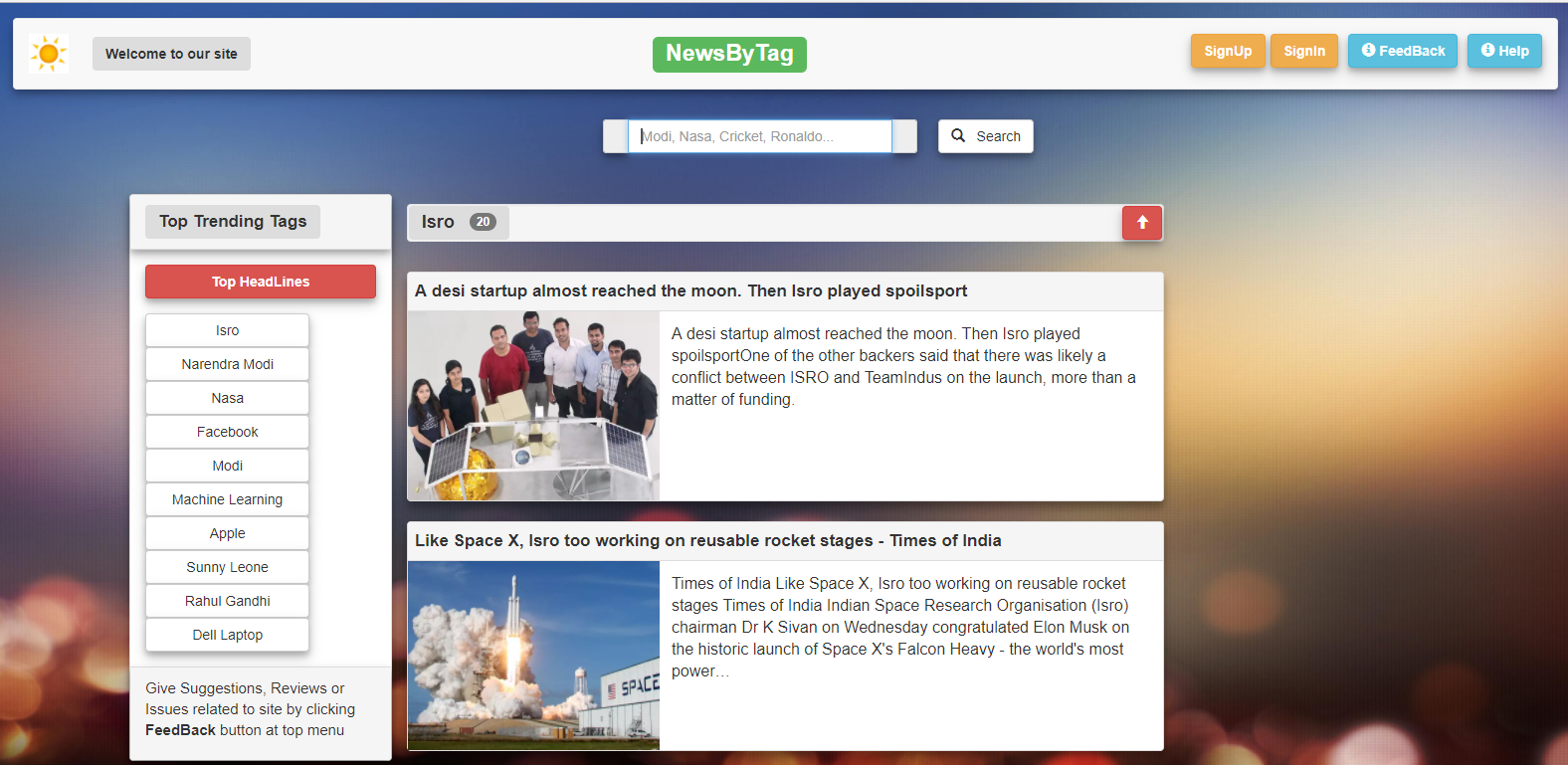
****

**Changing Password Page**

****

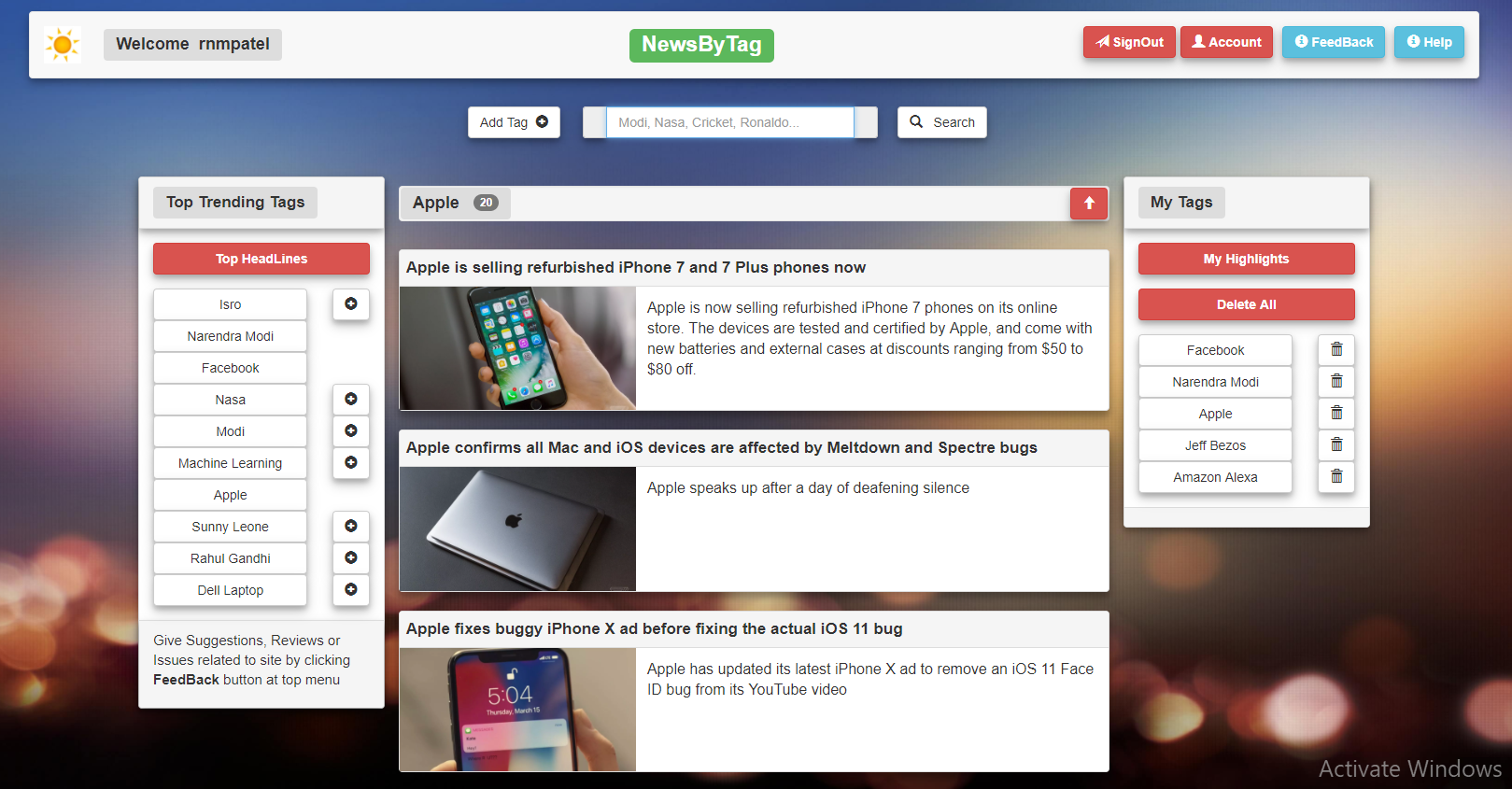
**Feedback Page**

**6.2 Home Page Before Sign-In**

****

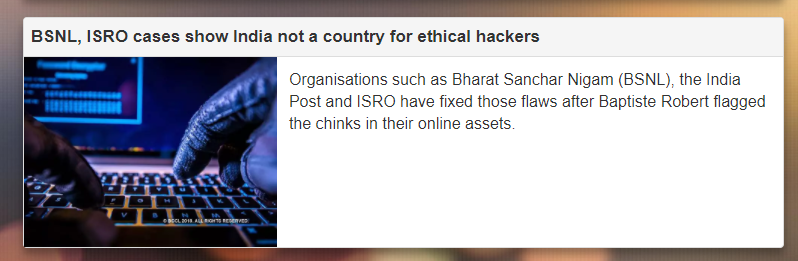
**Home Page Before Sign-In**

**6.3 Home Page After Sign-In**

****

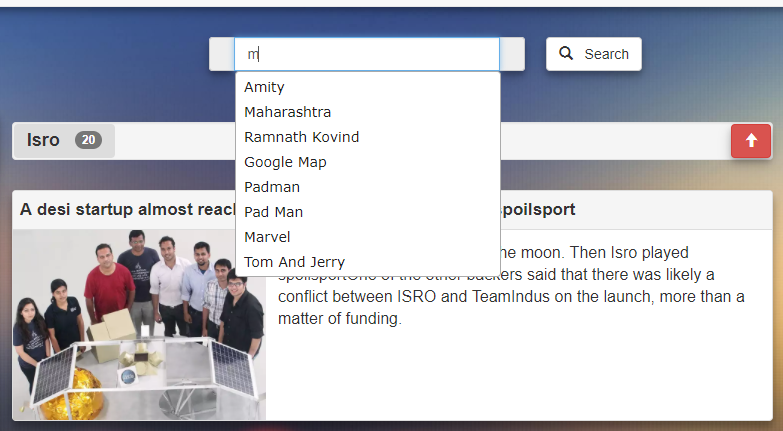
**Home Page After Sign-In**

**6.3 Close News View**

****

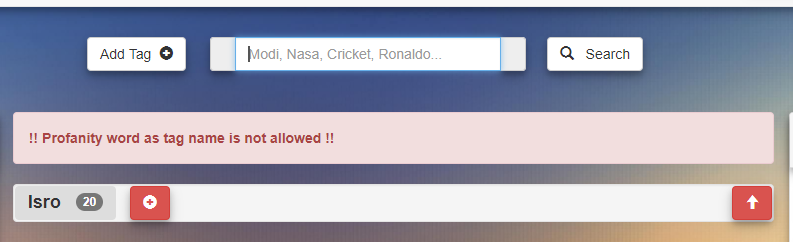
**Close news view**

**6.2 Autocomplete Search**

****

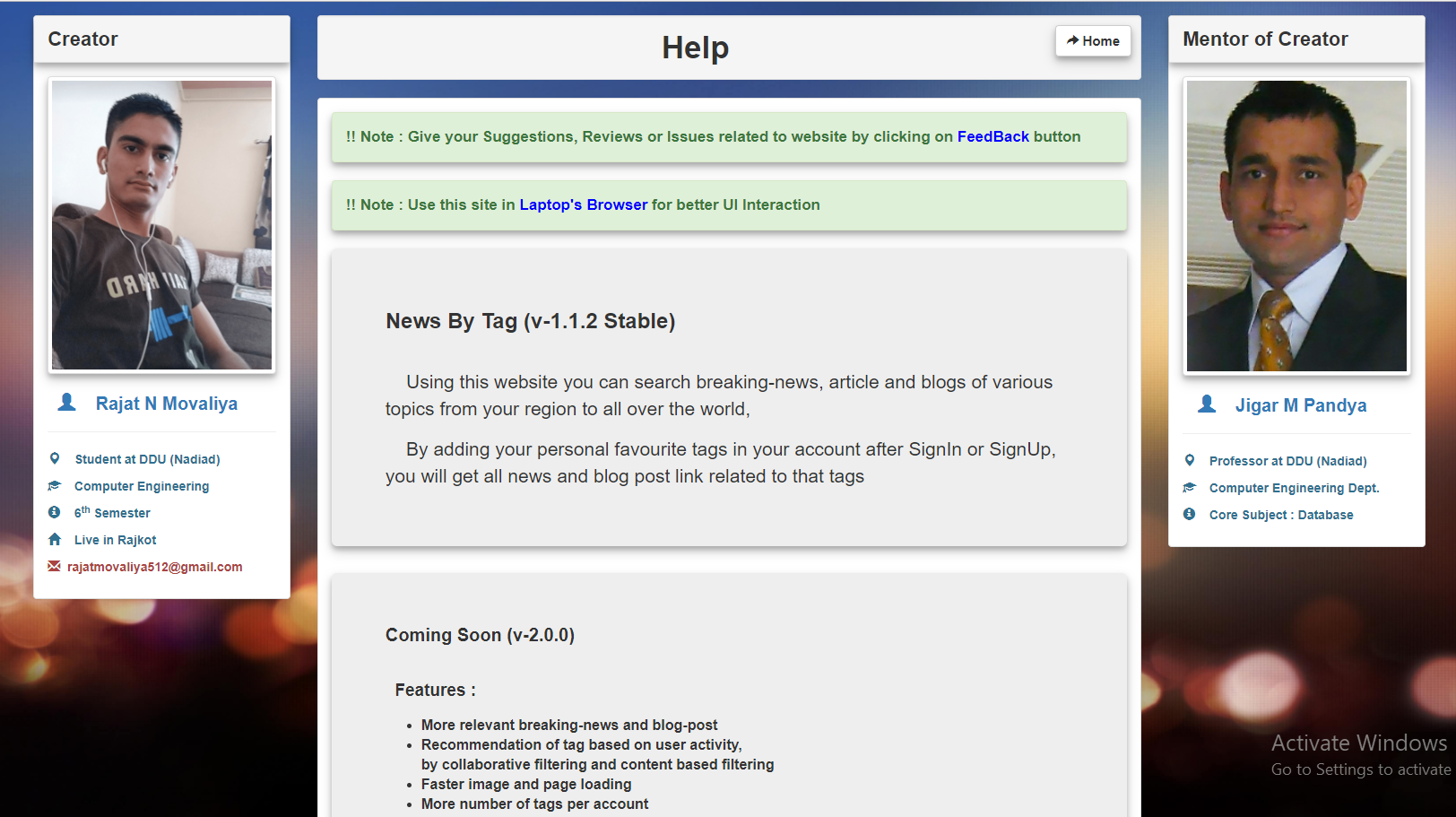
**Autocomplete search after entering some input**

**6.2 Profanity word as a tag name**

****

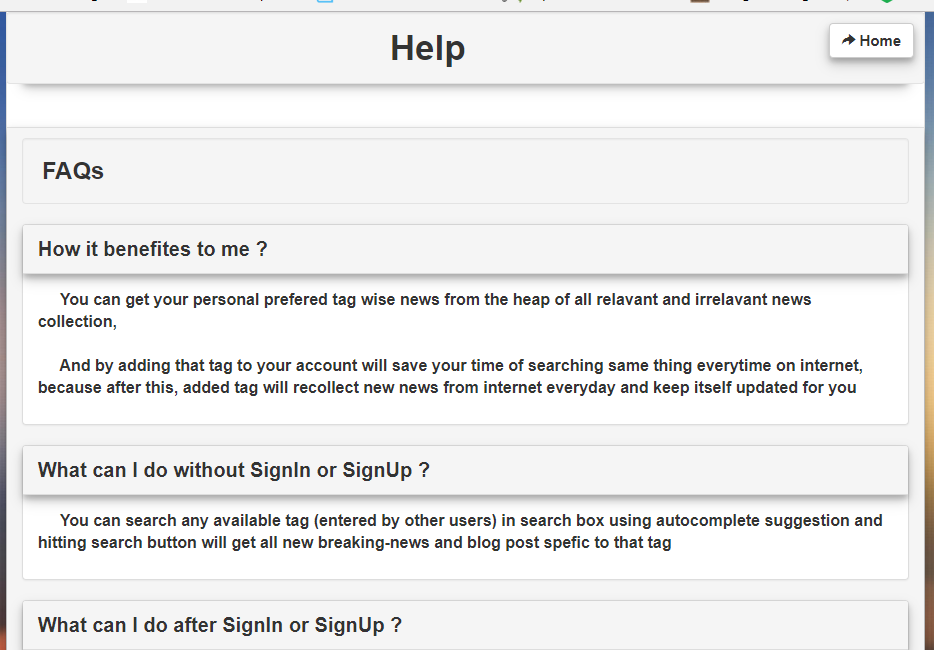
**Error message after trying to add profanity word as the tag name**

**6.5 Help Page Overview**

****

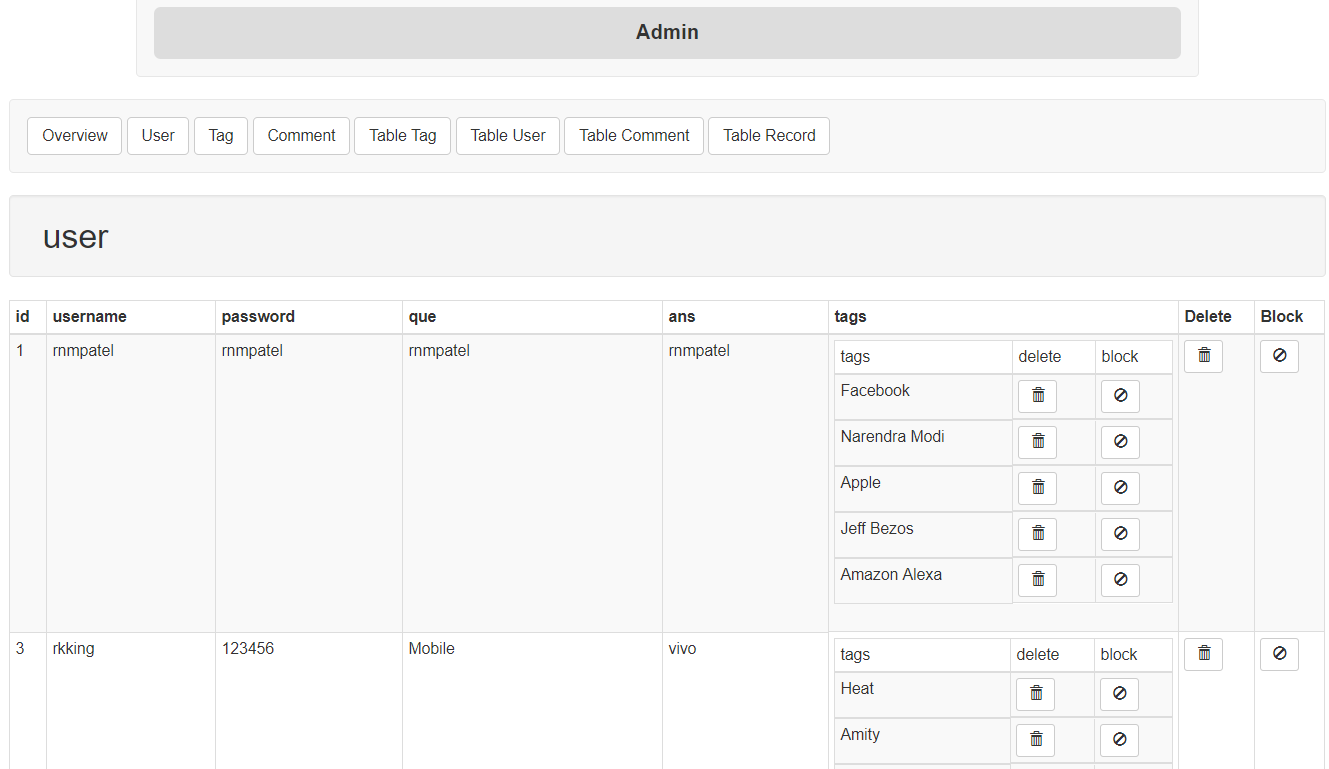
**Help page overview**

**6.5 Help Page Close view**

****

**Help page overview**

**6.5 Admin Panel**

****

**Admin Panel Page**

**CHAPTER: 7**

**Software Version Deployment Steps**

* **Software Version Deployment Steps**
* **Steps**
* Pre-Requirement : Linux Based OS, Postgresql Client for Database operation
* Open terminal in Ubuntu OS, and go to directory of Host Project run main file by typing command “python3 app.py” this will host web api hosting program in 5000 port with ip: 127.0.0.1 (localhost) with it’s own built in development web server.
* Go to Consumer Program and run following command “python3 app.py”, this will host consumer program in 5001 port with ip:127.0.0.1 (localhost).
* Now open any web browser and enter url “localhost:5001” will fetch index page from web app and continue using it.

**CHAPTER: 8**

**LIMITATIONS**

**AND**

**FUTURE ENHANCEMENTS**

**8. Limitations and Future Enhancements**

* **Limitations**

1. Admin needs to manually add vulgar tags entered by the user to block list, in case system does not recognize it.
2. Many rare tags do not have that frequent new breaking news, so the user has to wait for a long time to news updates for that tag.

* **Future Enhancements**

1. Recommendation of tags based on user activity(collaborative filtering and content-based filtering)
2. More and more tag relevant news and blog post.
3. Android App can be made because it makes a lot easier for the user to use with offline news storage facility and reduce the number of steps to open news
4. Favorite news storage facility.
5. News in other than English language as well
6. Filtration of news based on post dates, the word in body and title, and much more.
7. The more Fastest process of fetching new tag news.
8. Loading images and pages faster by CDN (Content Delivery Network) and by providing the various size of thumbnail versions of images.
9. Adding SSL (Secure Socket Layer) based communication to add more security to credentials.

**CHAPTER: 9**

**CONCLUSION**

**9. Conclusion**

Hereby, we declare that the functionality implemented in this system was performed by understanding all the modules.

All functionalities in the SRS are successfully implemented as described.

This project aims to provide breaking news by tag names provided by the user which was fulfilled to a greater extent. All the diagrams were prepared before implementation and the coding was done accordingly. After the coding was completed, comprehensive testing was performed and the results were provided in the report. Unit Testing of all modules was done and later, Integration Testing was also performed.

**CHAPTER: 10**

**BIBLIOGRAPHY**

1. **Bibliography**

**Web Tutorials:**

* <http://docs.sqlalchemy.org>
* http://flask.pocoo.org/docs/0.12/tutorial
* <https://www.tutorialspoint.com/python/index.htm>
* <https://www.tutorialspoint.com/postgresql/index.htm>
* <http://www.vogella.com/tutorials/Git/article.html>
* <https://devcenter.heroku.com/categories/reference>
* <http://jinja.pocoo.org/>