

Chapter 1

Introduction

1.1 Background

Question and Answer (Q&A) systems play an important role in our way of life for information and data sharing. Users post queries and choose inquiries to answer within the system. Thanks to the speedily growing user population and therefore the range of queries, it's unlikely for a user to come across a matter by accident that he/she will answer. Also, selflessness doesn't encourage all users to produce answers, to not mention top quality answers with a brief answer wait time. For this, RETORT has been designed and enforced, a web social network primarily based Q&A system. RETORT uses to its advantages the social network properties of common-interest associate degreed mutual-trust friend relationship to spot an utterer through relationship who will possibly answer the question, and enhance the user security. RETORT has been conjointly improved with security and potency enhancements by protective user privacy and identifies, and retrieving answers mechanically for perennial queries. The results counsel that social networks will be leveraged to boost the solution quality and asker's waiting time.

1.2 Problem Statement

The Internet is a significant wellspring of data, where the measure of information is tremendous and continually developing. Clients depend on web search tools to discover explicit data in this information base. Web crawlers, for example, Google and Bing use catchphrases given by the clients to perform looks. As of late, mechanical innovative work exercises, for example, Microsoft and Facebook's social-highlighted Bing look attempt, endeavor to join web crawlers and online informal organizations for higher pursuit execution. As past research has shown, web indexes perform well in ordering website pages and furnishing clients with applicable substance to their look however are not appropriate for

non-genuine inquiries, for example, "Which is the best nearby auto shop?". To address this specific class of non-authentic inquiries, many Question and Answer (Q&A) frameworks, for example, Yahoo! Answers, Baidu Zhidao, StackExchange, Quora and Ask have been created. Since their beginning, Q&A frameworks have demonstrated to be an important asset for sharing skill and thus are utilized by an enormous number of Internet clients. For instance, Yahoo! Answers was propelled toward the year's end 2005 and pulled in excess of 10 million clients in February , and hit 200 million clients in December of 2009. Questions and answers frameworks additionally save all inquiries and answers, in this manner going about as a vault for data recovery. They are significant for sharing specialized information, yet in addition as a hotspot for accepting exhortation and fulfilling one's interest about a wide assortment of subjects. Accordingly, current Q&A frameworks may not meet the prerequisite of furnishing top notch answer with a short answer hold up time, however clients wish to get acceptable answers rapidly. This is affirmed by the examination in. It found that for Yahoo! Answers, just 17.6% of inquiries were addressed attractively; for the staying 82.4%, one fifth of the inquiries stayed unanswered. For Baidu Zhidao, 22.7% of inquiries were effectively replied, and 42.8% of the uncertain inquiries were not replied by any means. In this way, there is an expanding requirement for a progressed Q&A framework that can diminish the quantity of unanswered inquiries, upgrade the appropriate response quality and lessening the reaction time.

1.3 Existing System

The developing significance of Q&A frameworks requests a push to all the more likely comprehend these frameworks and to improve. The works in concentrated the impact of various variables (e.g., clients' profiles, messages forecast, framework collaborations and network estimate) in the informal organizations on Q&A execution. These examination results establish the framework of RETORT to use interpersonal organization properties in the plan. Note that the current informal community dependent on the asker-answerer relationship in current Q&A frameworks is unique in relation to online interpersonal organization dependent on the social relationship, which is utilized in RETORT. The works

in focused on finding specialists and definitive clients. Rather, RETORT means to discover ordinary clients that can respond to questions including sentiment type questions. A few examinations have been led to make notoriety models in Q&A frameworks to build the believability of answers, and to decide the connection between the notoriety of the clients and the nature of their gave answers .RETORT legitimately uses the interpersonal organization property of common trust companionship to persuade clients to give answers without depending on an extra notoriety model. RETORT imparts closeness to other companion aide frameworks, for example, in utilizing the aggregate intensity of friends for a specific objective. Some exploration orders inquiries into predefined classifications, making it simpler for clients to find recently posed inquiries and for specialists to discover addresses they can reply. Quan et al. proposed three new regulated term weighting plans for inquiry classification, and assessed each plan utilizing a follow from Yahoo! Answers. Melody et al. proposed a successive procedure including point insightful word distinguishing proof and weighting, semantic mapping, and closeness estimation.

1.4 Proposed System

We have created and prototyped an online informal community based Q&A framework, called RETORT. It uses the properties of an interpersonal organization to advance an inquiry to potential answer suppliers, guaranteeing that a given inquiry gets a great answer in a brief timeframe. It expels the weight from answer suppliers by straightforwardly conveying them the inquiries they may be keen on, rather than requiring answer suppliers to look through an enormous accumulation of inquiries as in Yahoo! Answers or flooding an inquiry to the majority of an asker's companions in an online interpersonal organization. The sprout channel based improvement techniques scramble the intrigue and companionship data traded between clients to secure client protection, and record all n-grams of addressed inquiries to naturally recover responds to for intermittent inquiry. The onion steering based answer sending ensures the personalities of askers and answers. Our exhaustive follow driven examinations and investigation results on this present reality Q&A exercises from the RETORT model demonstrate the guarantees of RETORT to upgrade answer quality and

lessen answer hold up time in current Q&A frameworks, and exhibit the protected and effectiveness improvement accomplished by the upgrades. Since same inquiries might be displayed in all respects diversely and a similar inquiry might be addressed contrastingly in various circumstance. Later on, we will participate with different methods (for example point displaying and word implanting) into RETORT to locate the excess inquiry with an enormous scale client set.

1.5 Objectives

The main objective RETORT is to boost the performance of Q&A systems by actively forwarding inquiries to users who are capable and want to answer the queries. RETORT also aims to reduce wait time for the user to get answer to their queries.

Chapter 2

Literature Survey

This chapter gives the literature survey of the project. The following research papers have been referred for the project.

2.1 SOS: A Distributed Mobile Q&A System Based on Social Networks

AUTHORS: Ze Li, Haiying Shen Jin Li, Guoxin Liu.

SOS melds an online casual network, where center points partner each other by their social associations. As showed up in Figure 1, an enrollment server is responsible for center point enrolment. Each customer has an interest ID, which addresses his/her favorable position. The closeness of two customer's bit of leeway IDs suggests the similarity between the two customers' interests. Customers bestowing progressively fundamental interests to an asker will undoubtedly have the choice to react to the asker's request. Moreover, customers having shorter social detachments with an asker will undoubtedly be glad to react to the asker's request. SOS has a metric similarity (S) that appraises the likelihood of a center point to be competent and willing to address another center point's request. It is constrained by the interest comparability between the request's preferred position and the recipient's eagerness similarly as the

Social closeness between the request recipient and sender. SOS portrays a consistent K, which is the greatest number of associates that a center point can send/forward a request in its sidekick list. SOS empowers each center point to describe TTL, which is the maximal number of bounces that a request can be sent. A center chooses TTL depending upon how basic the request is. Figure shows the request directing procedure in SOS. After asker A

begins a request, it progresses the request to the top K mates (center points B and C) who have the most raised S in its sidekick list with the request.

2.2 Pythia: A Privacy-enhanced Personalized Contextual Suggestion System for Tourism

AUTHORS: G. Drosatos, P. Efraimidis, A. Arampatzis, G. Stamatelatos, and I. Athanasiadis.

Pythia is another security updated customer driven sensible suggestion system for the movement business. The system manhandles the customer's progressed pursue to normally create and invigorate a profile, which is safely kept at the customer side. An applicable proposition part, executed at the customer's side, ponders the profile and a zone of interest, and delivers POI suggestions.

The system contains convenient, common (work region or server-side), and cloud sections. An individual circulated stockpiling fills in as widely appealing amassing for the application parts and a POI gathering structure manages the available POI data of the system.

2.3 Contact Analysis in Workplace

AUTHORS: Sabrina Gaito, Elena Pagani ,Gian Paolo Rossi.

The arrangement of a specific contraption for pursue recording is principally awakened by the need of watching and recording amazingly short contact periods, couple of minutes, that rise up out of self-assertive flexibility in a thick domain. As a result, PMTRs have been proposed to work with beaconing times running from 1 sec. to some configurable regard which depends upon the flexibility condition we wish to watch. In addition, the contraptions need to enable unmanned preliminaries suffering 3 every month without batteries substitution. Layer 2 guides are the uncommon housings a PMTR imparts to its neighbors. The primary go through a reference point is gotten from a given encounter, the here and now

is recorded for the contact start, together with the experience ID; a contact closes when signals from a particular encounter have been missing for more than t seconds, with $t = 60$ in our preliminaries. The close-by memory measure should be dimensioned to store the contacts of the preliminaries. Our demonstrating grounds have made all things considered 2000 contacts for every contraption with beaconing time set to 1 sec. No specific information transmission and taking care of necessities have been imagined for PMTRs.

2.4 Effects of expertise differences in synchronous social Q&A

AUTHORS: Matthew Richardson and Ryen W. White

IM-an-Expert is a computerized administration that gets questions through IM, finds and contacts potential answerers with expertise or enthusiasm for the inquiry theme, and intercedes the exchange among asker and answerer. The asker starts an IM discussion with IM-an-Expert and suggests the conversation starter. The inquiry is utilized to recover a candidate set of experts dependent on profile data (portrayed later). A little gathering of those experts who are at present accessible (not occupied, away, in a gathering, in a call, and so forth accessible by means of quality data from the IM customer) are reached through IM three at a time, in dropping request of their expertise, to decide if they are happy to help answer the inquiry. On the off chance that and when an answerer acknowledges, different solicitations are cancelled. On the off chance that a candidate answerer does not react in time or rejects the inquiry, the administration asks others. When an answerer acknowledges, IM-an-Expert intervenes the discussion between the asker and answerer. When the discussion closes, the asker is solicited to alternatively rate the quality from the answer they got on a scale from one (not accommodating) to five (exceptionally supportive). To help the usefulness depicted in this section, the framework has two parts: (I) expertise locator: chooses clients who are in all probability ready to answer an inquiry, and (ii) exchange manager: handles question preparing and correspondence management all through the inquiry lifecycle.

CHAPTER 3

System Requirements Specification

System requirements specification (SRS) is a nitty gritty detail of both the product and equipment segments required, which are important for the framework execution, alongside practical and non-useful and operational necessities, as foreseen from the framework.

Programming prerequisites detail sets up the reason for understanding among designers and the end client on what the product item is to do just as what it isn't required to do. Programming necessities determination allows a thorough appraisal of prerequisites before configuration can start and diminishes later upgrade, it ought to likewise give a sensible premise to assessing item costs, amount, and endowments.

The product prerequisites determination report enrolls enough and essential necessities that are required for undertaking improvement. To infer the necessities we need clear and exhaustive comprehension of the items to be created or being created

The prerequisite of venture includes the necessities that ought to be available to give the best outcome. It can have following subsections-Requirement Engineering, Types of necessities, Hardware and Software Requirement and Feasibility.

3.1 Requirement Engineering

Requirement engineering (RE) alludes to the way toward characterizing reporting and keeping up the prerequisites of framework designing and programming building worried about this procedure. In the cascade model, prerequisite designing is exhibited at the principal period of advancement process.

3.2 Types of Requirement

3.2.1 Functional Requirements

- **User Sign up** :-all the users have to enter all the mandatory fields and register an account in our application to access our application.
- **User Login** :- To access the application, it is an necessity to validate the users login user name and password.
- **Admin Login** : This is super user of the application where he/she can login into the application with standard user name and password
- **Queries** : After login as all users they can view the queries of all members.
- **My Queries** : User can post his query on queries and view all replies on his queries.
- **Friend Request** : User can find the friend requests and he can accept that request.
- **Friend Search**: User can search friends and send them a friend request.

3.2.2 Non-Functional Requirements

- **Expanded System admin security**: overseer to eschew the abuse of the application by PC ought to be exceptionally secured and available.
- **Compactness**: The Presentation of this application is facile to utilize so it is looks simple for the using client to comprehend and react to identically tantamount.
- **Unwavering quality**: The functionalities accessible in the application this substructure has high probability to convey us the required inquiries.
- **Time take for Reaction**: The time taken by the application to culminate an undertaking given by the client is very fast.
- **Multifariousness**: Our application can be stretched out to incorporate the vicissitudes done by applications present now to enhance the performance of the item. This is implicatively insinuated for the future works that will be done on the application.
- **Vigour**: The project is blame tolerant concerning illicit client/beneficiary sources of information. Blunder checking has been worked in the platforms to avert platforms disappointment.

3.3 System Requirement

3.3.1 Hardware requirements

The most generally perceived course of action of essentials described by any working structure or programming application is the physical PC resources, generally called hardware, the gear necessities list is as often as possible joined by a hardware closeness list, especially if there ought to be an event of working systems. A HCL records attempted, impeccable, and every so often incongruent hardware contraptions for a particular working system or application. The going with sub-fragments inspect the various pieces of hardware essentials. All PC working structures are proposed for a particular PC plan. Most programming applications are limited to explicit working systems running on explicit structures. Disregarding the way that building free working systems and applications exist, most ought to be recompiled to continue running on another plan.

The vitality of the central getting ready unit (CPU) is a focal structure need for any item. Most programming running on x86 building portray planning power as the model and the clock speed of the CPU. Various features of a CPU that sway its speed and power, like transport speed, store, and MIPS are often neglected. This importance of vitality is normally wrong, as AMD Intel Pentium CPUs at similar clock speed much of the time have particular throughput speeds.

.

Processor	Intel Core i5 or AMD FX 8 core series with clock speed of 2.4 GHz or above
RAM	2 GB or above
Hard disk	40 GB or above
Input device	Keyboard or mouse or compatible pointing devices
Display	XGA (1024*768 pixels) or higher resolution monitor with 32 bit color settings
Miscellaneous	USB Interface, Power adapter, etc.

Table 3.1: Hardware Specification

3.3.2 Software requirements

Programming necessities oversee portraying programming resource necessities and prerequisites that ought to be acquainted on a PC with give perfect working of an application. These necessities or prerequisites are generally barred in the item foundation pack and ought to be presented freely before the item is presented.

Operating System	Windows XP or above
Programming Language	Python (3.6)
Integrated development environment (IDE)	Visual Studio Code
Algorithm Used	Naïve Bayes Classifier

Table 3.2: Software Specification

3.4 Feasibility Study

Feasibility study is a significant stage in programming advancement process. It empowers the designers to have an appraisal of the item being created, as far as results of the item, its operational use and specialized help required for actualizing it. An investigation and assessment of a proposed venture to decide whether it is in fact plausible, is possible inside the evaluated expense, and will be gainful. Practicality ponders are quite often led where huge entireties are in question.

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

3.4.1 Technical Feasibility

The advancements required for your application is accessible in the market. Additionally the product can be actualized with existing technical assets. Technical feasibility is one of the principal ponders that must be directed after the venture has been recognized.

The Technical Feasibility Study computes materials required and evaluates the subtleties of how you will convey an item or administration. The Proposed System can be transferred to Cloud where every one of the calculations will occur. So it is a most ideal approach to contact all individuals since advanced mobile phones are so natural to deal with and even workstations are practically accessible to many.

3.4.2 Operational Feasibility

Operational feasibility is a proportion of how well a proposed framework takes care of the issues, and exploits the open doors recognized amid degree definition. The proposed application is easy to use with great GUI. The client just advances question to his companions, who has those interests. Henceforth, the created framework has operational plausibility.

3.4.3 Economical Feasibility

Economical Feasibility is utilized to decide the advantages and reserve funds that are normal from a proposed framework. Use acquired for building up the new framework will be practical or not. It is utilized to decide the advantages and reserve funds that are normal from proposed framework as this framework is open source it won't cost much. It requires just windows stage, for example workstations and PCs which are broadly utilized now days.

Chapter 4

System Analysis and Design

The dissection of a system into its component pieces to study how these component pieces interact and work is known as system design. System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements.

4.1 Overview of System Design

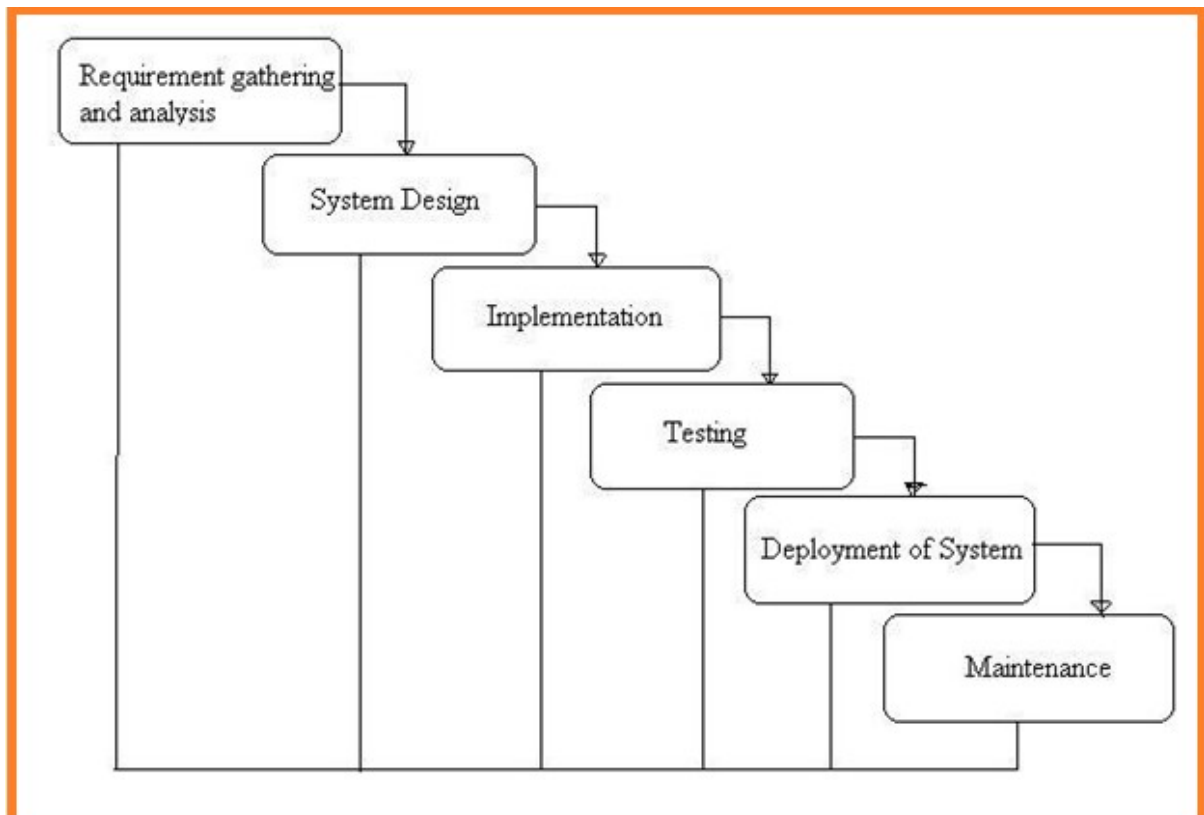


Figure 4.1: Overview of System Design

The above Figure 4.1 shows the overview of system design, central to which the RETORT has been made. Here, Windows is the platform which is connected to the backend server and database.

4.2 Methodology

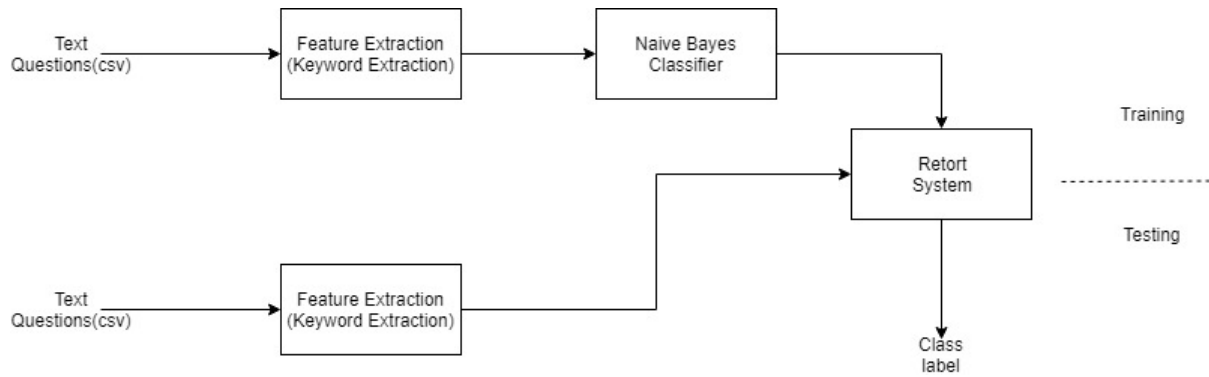


Figure 4.2: Methodology

4.3 UML Diagram

4.3.1 Class Diagram

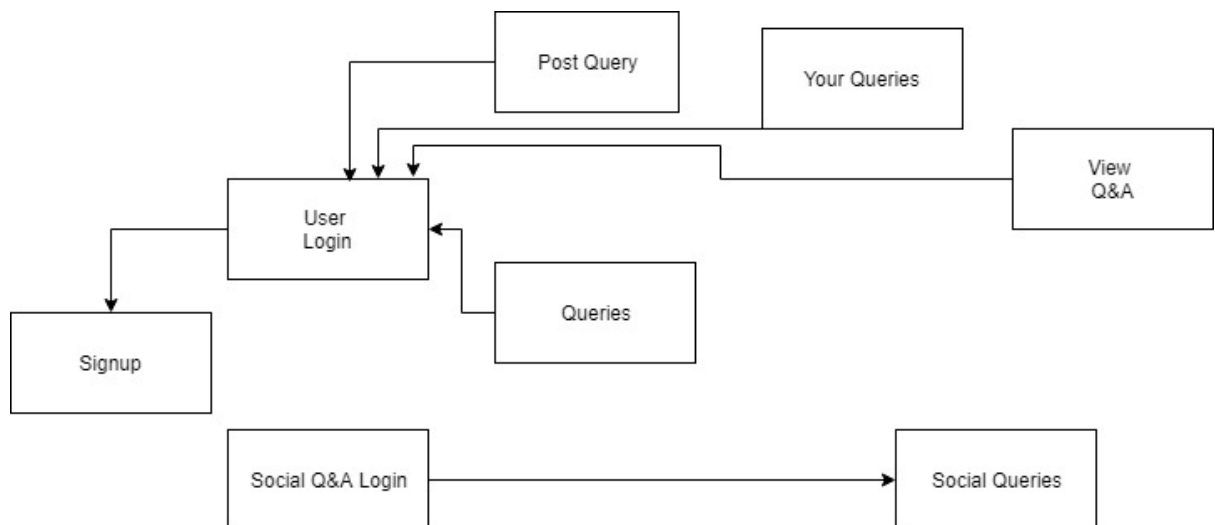


Figure 4.3: Class Diagram

4.3.2 Use Case Diagram

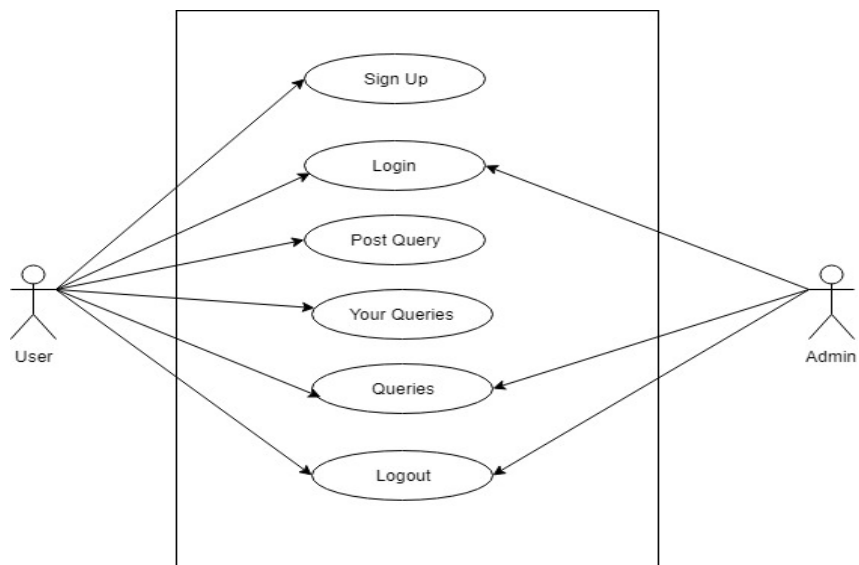


Figure 4.4: Use Case Diagram

4.3.3 Data Flow Diagram

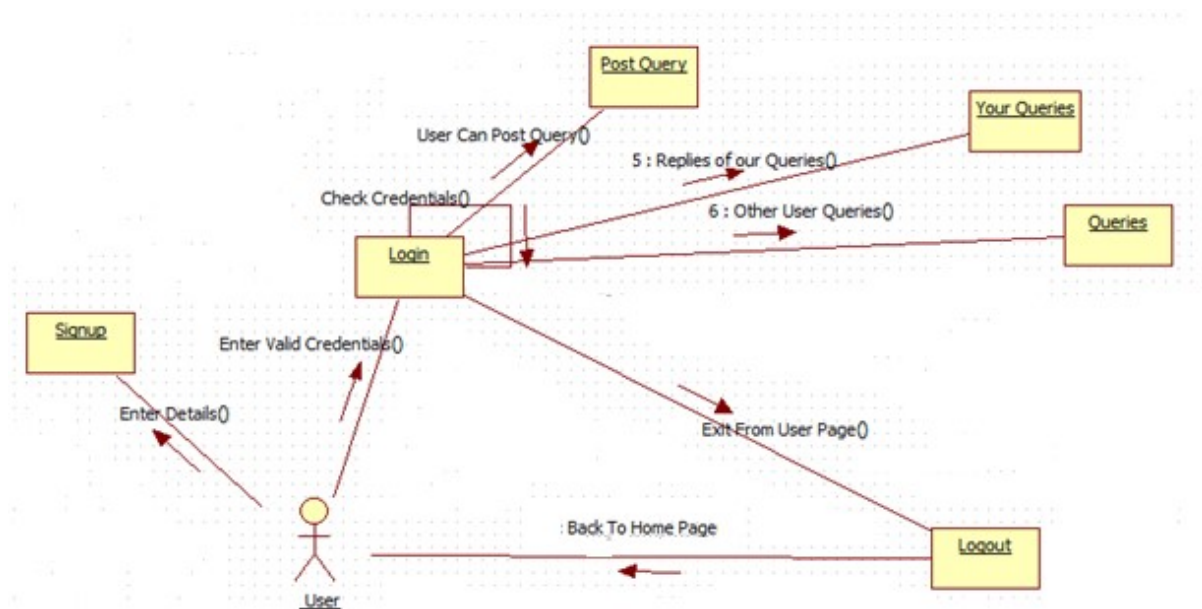


Figure 4.5: User Data Flow Diagram

4.3.4 Sequence Diagram

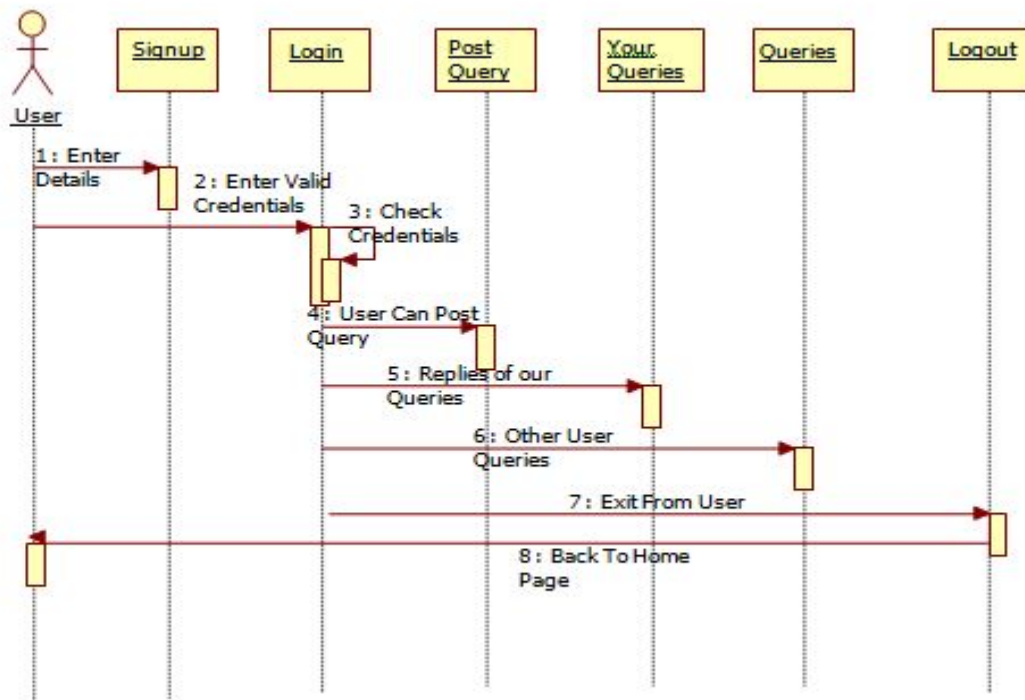


Figure 4.6: User Sequence Diagram

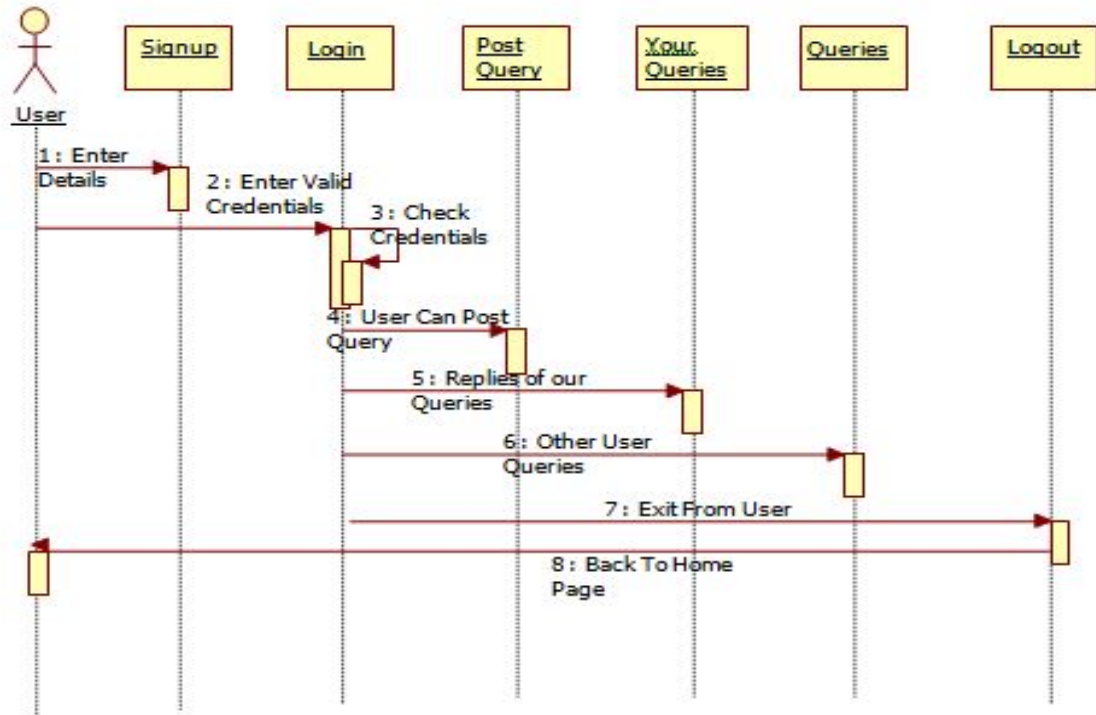


Figure 4.7: Admin Sequence Diagram

4.3.5 Component Diagram

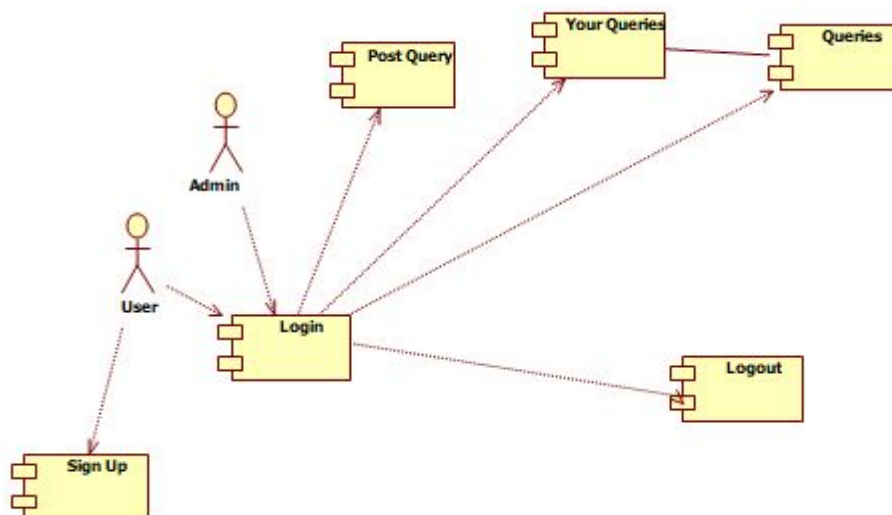


Figure 4.8: Component Diagram

Chapter 5

Implementation

Implementation is the procedure that transforms methodologies and plans into activities so as to achieve vital destinations and objectives. Actualizing your vital arrangement is as significant, or much progressively significant, than your procedure.

5.1 Modules

5.1.1 User Module:

Client can login as register qualifications. After login as client can look companions, send as companion solicitation and discover the companions who send the companion demand. Client can post the inquiries, see the answers of on posted questions and answer the some other companion inquiries.

5.1.2 Admin Module:

The Admin recognizes the inquiries classification and add the inquiry to the class. He/She can see all answers of questions. After then send inquiries to related intrigue clients to address that question. Ascertain the appropriate responses according to calculation and appoint the qualities.

5.2 Algorithm used in Project

In AI we are regularly keen on choosing the best hypothesis (h) given information (d).

In a characterization issue, our hypothesis (h) might be the class to relegate for another information example (d).

One of the least demanding methods for choosing the most likely hypothesis given the information that we have that we can use as our prior learning about the issue. Bayes'

RETORT

Theorem gives a way that we can figure the probability of a hypothesis given our prior information.

Bayes' Theorem is expressed as:

$$P(h|d) = (P(d|h) * P(h))/P(d)$$

Where

- $P(h|d)$ is the probability of hypothesis h given the information d . This is known as the back probability.
- $P(d|h)$ is the probability of information d given that the hypothesis h was valid.
- $P(h)$ is the probability of hypothesis h being valid (paying little mind to the information). This is known as the prior probability of h .
- $P(d)$ is the probability of the information (paying little mind to the hypothesis).

You can see that we are keen on ascertaining the back probability of $P(h|d)$ from the prior probability $p(h)$ with $P(D)$ and $P(d|h)$.

In the wake of ascertaining the back probability for various theories, you can choose the hypothesis with the most elevated probability. This is the most extreme plausible hypothesis and may formally be known as the greatest a posteriori (MAP) hypothesis.

This can be composed as:

$$\text{MAP}(h) = \max(P(h|d))$$

or on the other hand

$$\text{MAP}(h) = \max((P(d|h) * P(h))/P(d))$$

or on the other hand

$$\text{MAP}(h) = \max(P(d|h) * P(h))$$

The $P(d)$ is a normalizing term which enables us to figure the probability. We can drop it when we are keen on the most plausible hypothesis as it is consistent and just used to standardize.

Back to characterization, in the event that we have a much number of cases in each class in our preparation information, at that point the probability of each class (for example $P(h)$) will be equivalent. Once more, this would be a steady term in our condition and we could drop it so we end up with:

$$\text{MAP}(h) = \max(P(d|h))$$

Naive Bayes is an order calculation for double (two-class) and multi-class characterization issues. The system is most straightforward to comprehend when depicted utilizing paired or all out info esteems.

It is called naive Bayes or numbskull Bayes in light of the fact that the estimation of the probabilities for every hypothesis are streamlined to make their figuring tractable. As opposed to endeavoring to ascertain the estimations of each characteristic esteem $P(d_1, d_2, d_3|h)$, they are thought to be restrictively autonomous given the objective esteem and determined as $P(d_1|h) * P(d_2|H)$, etc.

This is an exceptionally solid supposition that is most improbable in genuine information, for example that the characteristics don't collaborate. By and by, the methodology performs shockingly well on information where this presumption does not hold.

5.3 Packages Used

5.3.1 Scikit-Learn

Scikit-learn was at first created by David Cournapeau as a Google summer of code venture in 2007.

Later Matthieu Brucher joined the venture and began to utilize it as separated of his proposal work. In 2010 INRIA got included and the primary open discharge (v0.1 beta) was distributed in late January 2010.

The task currently has in excess of 30 dynamic donors and has had paid sponsorship from INRIA, Google, Tinyclues and the Python Software Foundation.

Scikit-learn creates a scope of administered and unsupervised learning calculations by means of a steady interface in Python. It is authorized under a lenient improved BSD permit and is dispersed under numerous Linux disseminations, empowering scholarly and business use. The library is based upon the SciPy (Scientific Python) that must be introduced before you can utilize scikit-learn. This stack incorporates:

- **SciPy**: Fundamental library for scientific computing
- **IPython**: Enhanced interactive console
- **Matplotlib**: Comprehensive 2D/3D plotting
- **Sympy**: Symbolic mathematics
- **Pandas**: Data structures and analysis
- **NumPy**: Base n-dimensional array package

Augmentations or modules for SciPy care traditionally named SciKits. All things considered, the module gives learning calculations and is named scikit-learn.

The vision for the library is a dimension of power and bolster required for use underway frameworks. This implies a profound spotlight on concerns, for example, simple of utilization, code quality, cooperation, documentation and execution.

Despite the fact that the interface is Python, c-libraries are influence for execution, for example, numpy for exhibits and network activities, LAPACK, LibSVM and the cautious utilization of cython.

5.3.2 Flask

Flask is a web system. This implies flask gives you instruments, libraries and innovations that enable you to manufacture a web application. This web application can be some site pages, a blog, a wiki or go as large as an online timetable application or a business site.

Flask is a piece of the classifications of the smaller scale structure. Miniaturized scale system are typically structure with practically no conditions to outside libraries. This has advantages and disadvantages. Professionals would be that the system is light, there are little reliance to refresh and look for security bugs, cons is that some time you should accomplish more work without anyone else or increment yourself the rundown of conditions by including modules. On account of Flask, its conditions are:

- Werkzeug a WSGI utility library
- jinja2 which is its format motor

In application improvement, a "model" alludes to the information portrayal of some genuine or theoretical item. For instance, in case you're constructing an application for a vehicle vendor, you may characterize a Car model that typifies the majority of a vehicle's qualities and conduct.

For this situation, you're fabricating a To-Do List with Tasks, and each Task has a place with a User. Before you ponder how they're identified with one another, begin by characterizing objects for Tasks and Users.

RETORT

The flask-sql alchemy bundle influences SQL Alchemy to set up and educate the database structure. You'll characterize a model that will live in the database by acquiring from the db.Model object and characterize the qualities of those models as db.Column examples. For every segment, you should determine an information type, so you'll pass that information type into the call to db.Column as the main contention.

Since the model definition consumes an unexpected theoretical space in comparison to the application setup, make models.py to hold model definitions separate from app.py. The Task model ought to be developed to have the accompanying properties:

- id: an esteem that is a novel identifier to pull from the database
- name: the name or title of the assignment that the client will see when the undertaking is recorded
- note: any additional remarks that an individual should need to leave with their errand
- creation_date: the date and time the undertaking was made
- due_date: the date and time the undertaking is expected to be finished (if by any stretch of the imagination)
- completed: an approach to show whether the undertaking has been finished

Chapter 6

System Testing

Introduction

The main purpose behind any testing procedure is to discover errors. Testing is the process of trying to discover every possible fault or point of weakness in a work product. The functionality of various components, sub-components, assemblies and a finished product can be checked using this process. It is the process of exercising with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. The system has been verified and tested by running the test data and live data.

6.1 Types of tests

6.1.1 Unit Testing

The first step towards any test process involves the design of test cases in order to validate the internal program logic, and also to check whether the program inputs produce valid outputs. All decision branches and internal code flow should be validated. Here the individual software units are checked for validation, which is usually done after the completion of an individual unit and before the process of integration. This is an example of spiral testing which mainly relies on the knowledge of its construction and is invasive. They are mainly performed at compound level and is responsible for testing a specific business process, application or system duration. It ensures that each unique path of a business process performs accurately documented specifications and contains clearly defined inputs and expected results.

Testing plays a major role in the software life cycle phase where it is a part of a combined code and unit test phase, although it is not uncommon for coding and unit testing to be conducted as two distinct events.

Strategy and approach

Testing is a process which is performed manually and functional tests will be written in detail.

- **Objectives**

- a. The field entries used in the process must work properly.
- b. Pages associated with an identified link related with the process must be activated.
- c. The entry screen, messages and responses to the process must not be delayed.

Cases to be tested

- a. All the entries must be verified so that they are in the correct format.
- b. Duplicate entries should not be allowed.
- c. It should always lead the user to the correct page.

6.1.2 Unit test considerations

In order to ensure that information flows properly into and out of the unit the module interface is tested. In order to maintain the integrity of all the steps during the execution of the algorithm the local data structures are examined periodically.

In order to ensure that module operates within the boundary bounded, the conditions are tested. All 'independent paths' through the control structures are restricted to ensure that all statements in a module have been executed at least once. At the end 'one-handling paths' are tested.

6.1.3 Unit test procedures

Testing is one of the procedures that is carried out during coding. Initially the source level code is tested, it reviewed by programmers and verified for correct syntax, it's after this process that unit test case design begins and a stand-alone program/driver is developed for

each test case. The main role of the driver is that it accepts test case data, which is tested with each module and finally prints the relevant results.

Tests are also done in order to replace modules called by the subroutines that is to be tested. A dummy subprogram uses the interface of the subordinate module, manipulates data with respect to it, and prints an entry for each verification and then returns. The drivers and scrubs are overhead i.e. both are software that must be written but that is not delivered with the final product. In order to reduce the overhead, the drivers and stub are kept low. The unit test procedure carried out in this project was found to be successful. Data flows evenly to all parts of the project.

6.1.4 Integration testing

Integration testing are designed and is mainly carried out to check if the integration of various software components builds into a standalone program. Testing is mostly driven by events and is more concerned with the basic outcome of fields. Integration tests is carried out to demonstrate that although the individual components are tested and are working properly, the combination too works correctly. Integration testing is responsible for recording the problems that arise from the integration of components.

Integration testing over here is the incremental integration testing of two or more software components on a single platform in order to detect the failures caused by interface defects. Integration test is mainly aimed to check whethereach components, e.g. components of software system or - one step up – software applications at the company level – interact with error.

Results: All the test cases mentioned above passed successfully. No defect encountered.

Integration testing is the phase of software testing where individual modules are grouped together and then tested. The purpose of integration testing is to verify functional, maintenance and reliability requirements placed on major design items. The design items mentioned above are accessed through their interfaces using a black box. Appropriate

parameters and data inputs are used to simulate success and error cases. Limited usage of shared data areas and inter-process communication is tested and individual items are exercised through their input interface. In order to ensure that all assemblages interact correctly, test cases are properly prepared, for example across procedure calls or process calls, and this is done after testing individual modules, i.e. unit testing.

The overall idea is a ‘building block’ approach, in which verified assemblages are added to a database which is then used to support the integration testing further assemblages. In this process, most of the modules that are developed are coupled together in order to form one of the major part of the system, which is used for integration testing. Integration testing is a systematic technique for constructing the program structure while simultaneously conducting test to uncover errors associated with interfacing. The main objective of unit testing is to use individual unit tested modules in order to build a structure which is a replica of the design.

6.1.5 Top-down Integration

The method is considered to be incremental approach followed for the construction of program structure. The integration of modules takes place by moving down through the control hierarchy which begins with the main program module. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. Integration testing is mainly carried out to check if the interaction of various components are carried out without an error.

Integration testing for Server Synchronization:

- a. Initially the IP address has to be tested which acts as medium for communication with other nodes.
- b. The route status in the cache table is checked after each node receives status information.
- c. The messages are displayed at the end of the application.

6.1.6 Testing Objectives

The objectives of testing includes the following:

- a. Testing is a process where a program is executed with the intention of finding an error.
- b. A good test case is one that has a high probability of finding an undiscovered error.
- c. A successful test is one that successfully uncovers an undiscovered error.

In order to uncover errors it is mostly recommended to conduct testing in accordance to the above mentioned objectives. It also ensures that each software function works according to the specification whose conditions are met by performance requirements.

6.2 Functional testing

Functional tests are mainly done in order to provide a demonstration that functions tested are available as specified by the business and technical requirements, system documentation and user manuals.

Functional testing is centered on the following items:

- | | | |
|--------------------|---|---|
| Valid Input | : | All classes which have valid input must be accepted. |
| Invalid Input | : | All classes having invalid input must be rejected. |
| Functions | : | All the identified functions must be exercised. |
| Output | : | All the identified classes of the application output must be exercised. |
| Systems/Procedures | : | All parts of the procedures and interfacing systems must be invoked. |

6.3 System testing

System testing ensures that all the requirements are met by the entire integrated software system. The system configuration is tested in order to ensure predictable results. It tests a configuration to ensure known and predictable results. Configuration oriented system test is one such example of system testing. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. System testing on any particular hardware or software, is a testing which is carried out on the entire system as a whole in order to evaluate the system compliance with the specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing takes as its input, all the integrated software components that have passed integration testing and also the software system integrated with any hardware system. Integration testing is mainly done in order to find out any inconsistencies between the software units or between any of the assemblages and the hardware. System testing is a more limited type of testing which means it detects both defects within inter-assemblages and also the system as a whole.

There are a variety of tests which are conducted on the system in order to explore functionality or to identify problems. It is usually carried out before or after the integration of the whole system. In order to perform system testing a series of systematic procedures are followed. The performance of the system and the common mistakes can be found out with the help of these procedures. Testers usually try to ‘break the system’ by entering data that may cause the system to malfunction or return incorrect information. For example, in order to check for an incorrect input from the user, a tester may put in a city in a search engine designed to only accept states, and checks the response of the system with respect to the input.

System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) and a System Requirement Specification (SRS). System testing tests are aimed not only to check the design of the system, but it also checks the behavior and the

believed expectations of the customer. System testing is done to test even beyond the bounds defined in the software or hardware requirements specification.

Name of the Test	System Testing
Item being tested	Over all functioning of application with all functions properly linked.
Sample Input	Sample input for each of the task provided out of the box.
Expected Output	Students get their respected tutors
Actual Output	Application reacts to user inputs in expected manner.
Remarks	Successful

Table 6.1: Test cases for Input-Output

6.3.1 White Box Testing

This is a testing procedure carried out in which the tester has complete knowledge about the inner workings, language and structure of the software. It is mainly used in those areas where black box testing cannot be implemented.

6.3.2 Black Box Testing

This is a testing procedure carried out in which the tester has no knowledge about the inner workings, language of the module and structure of the software. Black box tests are usually written from a definitive source document like specification or requirement document. It is a testing procedure where the software which is being tested is called the black box, which means we cannot see into it. It is basically intended to provide inputs and to respond to the outputs without bothering about how the software works.

6.3.3 Acceptance Testing

Acceptance testing is considered to be one of the most difficult phase of any project and it requires complete participation from the end user. Functional requirements of the system is given utmost importance and it ensured that these requirements are met.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3.4 Performance Testing

Performance testing ensures that the output is produces within the same time limit which includes the system time required for compiling, response sent to the users and the request sent to the system to retrieve the results.

6.3.5 Experimental Results

Dataset:

Class	Number of Samples
Movie	146
Food	235
Politics	34
Physics	102

Table 6.2 Samples in each class

RETORT

Training	Classification Accuracy	Precision	Recall	F measure
40%	0.7395	0.6089	0.5310	0.7395
50%	0.7526	0.6215	0.5454	0.7526
60%	0.7606	0.6292	0.5541	0.7606
70%	0.7674	0.6351	0.5608	0.7674
80%	0.7724	0.6382	0.5657	0.7724

Table 6.3: Accuracy of RETORT

Five different tests have been done on RETORT to check for its accuracy.

In the first test, the training size was 40% of the dataset and the testing size of 60% of the dataset. We achieved a classification accuracy of 73.95% in this case.

In the second test, the training size was 50% of the dataset and the testing size of 50% of the dataset. We achieved a classification accuracy of 75.26% in this case.

In the third test, the training size was 60% of the dataset and the testing size of 40% of the dataset. We achieved a classification accuracy of 76.06% in this case.

In the fourth test, the training size was 70% of the dataset and the testing size of 30% of the dataset. We achieved a classification accuracy of 76.74% in this case.

In the fifth test, the training size was 80% of the dataset and the testing size of 20% of the dataset. We achieved a classification accuracy of 77.24% in this case.

6.4 Test Cases

6.4.1 Test Cases for Register Page

Test Case 1		Test Case Description - Verification in Registration Form		
Sl.no.	Pre-requirements	Sl.no.	Testing Data Requirement	
1	User should be Recorded	1	Data ought to be substantial	
Test Condition				
Entering data in registration form				
Act no.	Act Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended
1	User enters First and Last Name	Popup indicating email confirmation message	Enter valid email/password	Failure
2	Submitting the form without entering any details	Popup indicating email confirmation message	Enter email /password	Failure
3	User enters invalid format of email id	Popup indicating email confirmation message	Enter valid email id	Failure

RETORT

4	User enters a phone number with < 10 digits	Popup indicating email confirmation message	Enter valid phone number	Failure
5	Entering valid username and password	Popup indicating email confirmation message	Pop showing email verification message	Passed

Table 6.4: Registration test case

6.4.2 Test Cases for Login Page

Test Case 2		Test Case Description - Verification in Login Form		
Sl.no.	Pre-requirements	Sl.no.	Test Data Requirement	
1	User should have an email id	1	Data ought to be substantial	
Test Condition				
Entering data in login form				
Step no.	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended
1	User gives a email or password of less than 6 characters	User has been logged in	Enter valid email/password	Failure
2	Submitting the form without entering any details	User has been logged in	Enter email /password	Failure
3	User enters wrong Email and (or) password	User has been logged in	Enter valid email /password	Failure

Table 6.5: Login test case

Chapter 7

Results and Discussion

In Figure 7.1, the login dashboard can be seen for the RETORT.

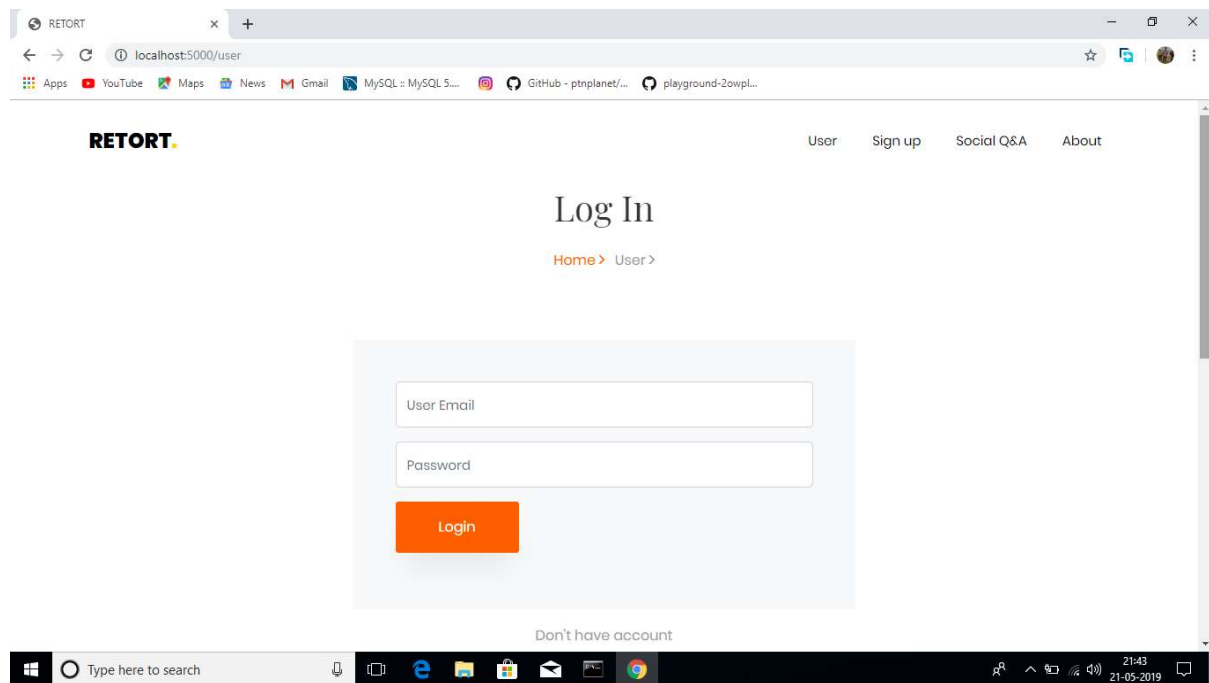


Figure 7.1: Login Page

RETORT

In Figure 7.2, the admin login can be seen for the RETORT.

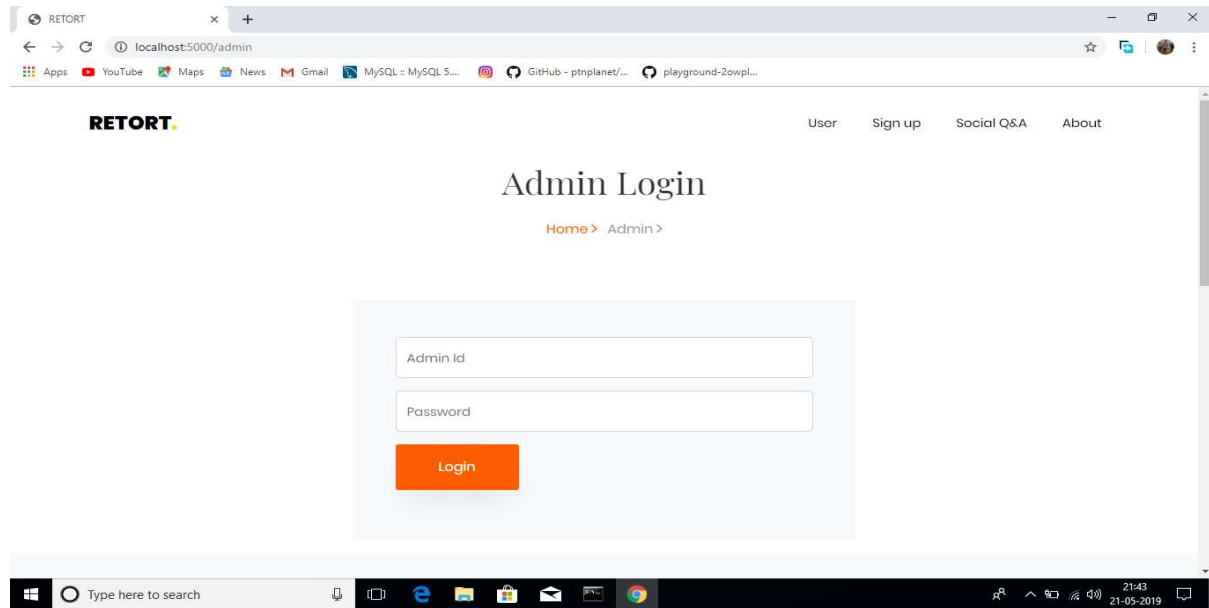


Figure 7.2: Admin Login

In figure 7.3, the signup menu for the new user can be seen, where various data like first name, last name, address, city etc. is collected from the user for successful signup.

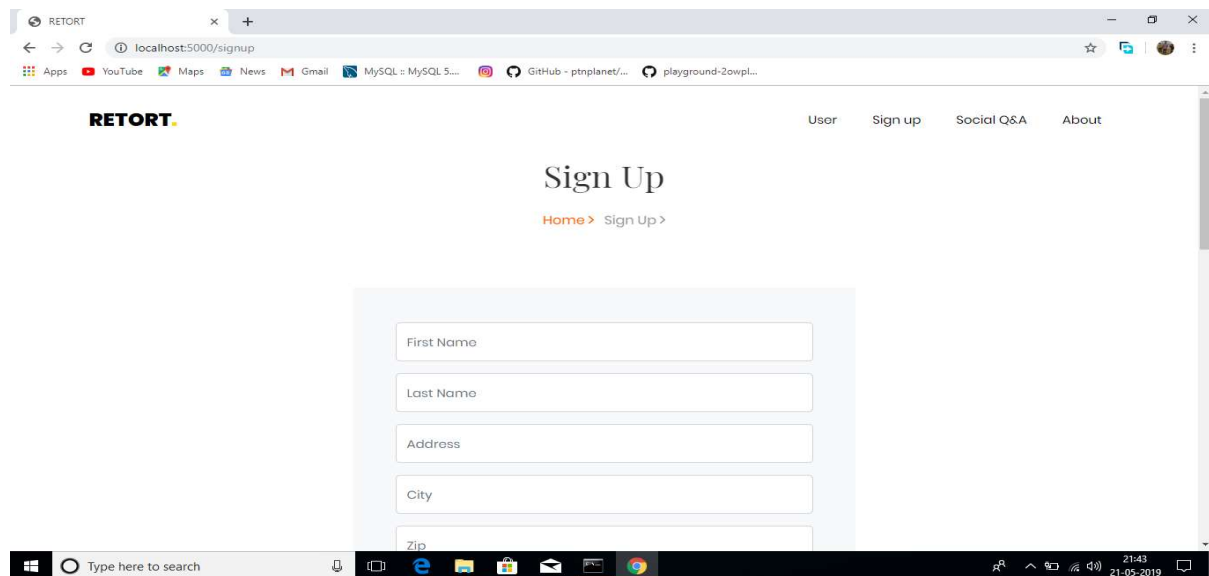


Figure 7.3: User Signup

RETORT

In Figure 7.4, the login dashboard can be seen of the user. Here, the user has various options like post query, answer other queries or look at his/her own queries asked.

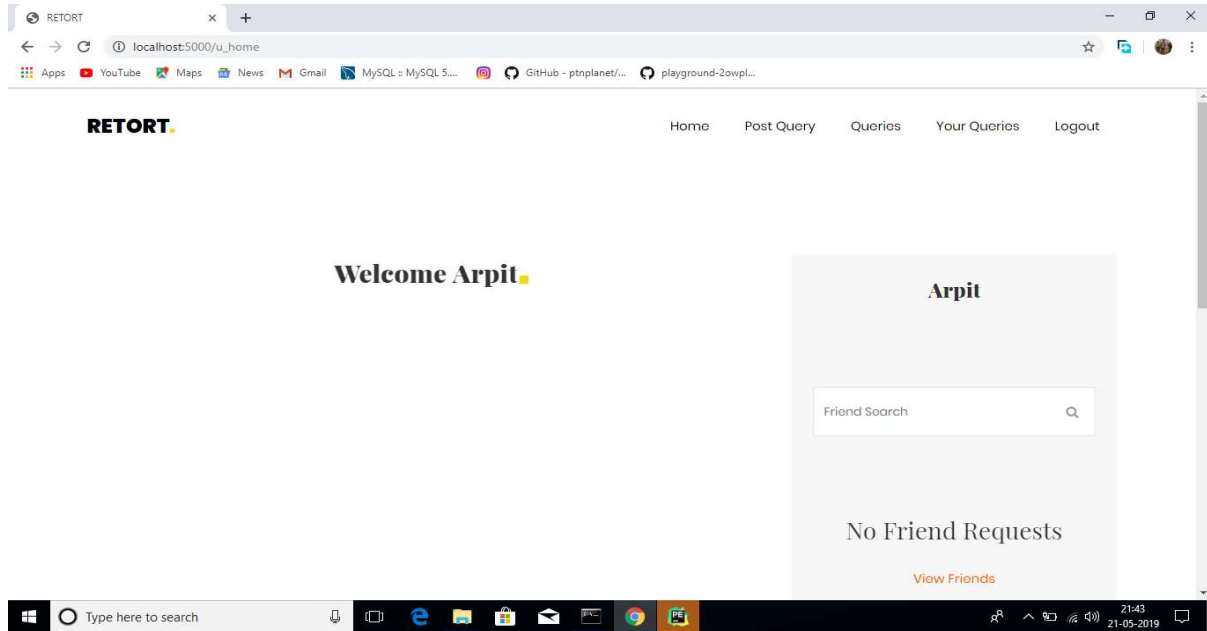


Figure 7.4: User dashboard

In Figure 7.5, we can see the user dashboard for posting a new query.

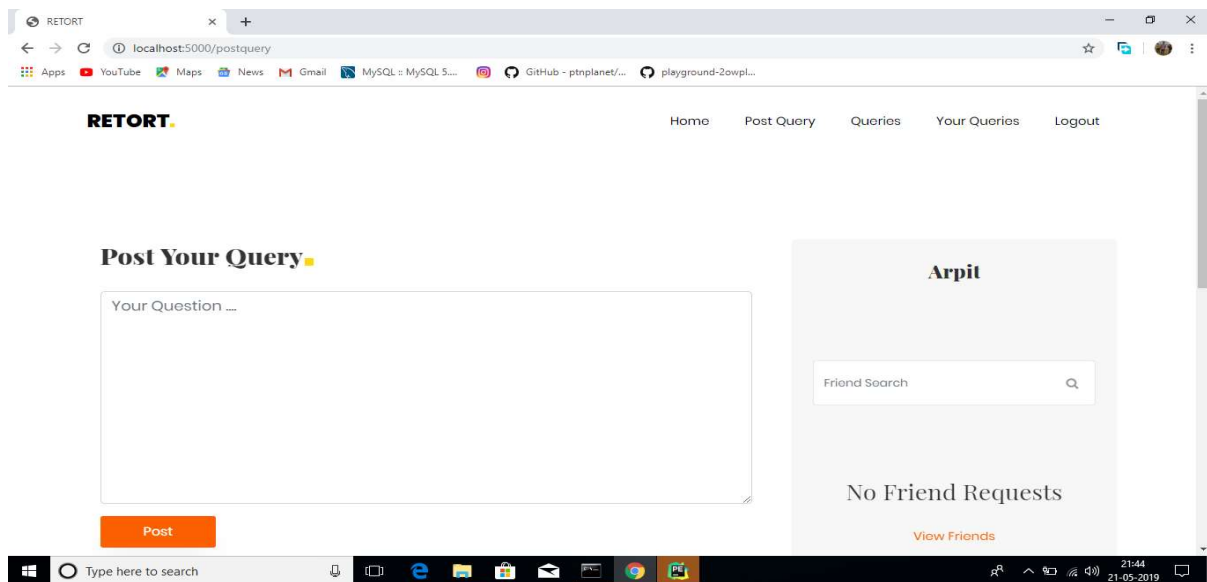


Figure 7.5: Post Query

RETORT

In Figure 7.6, we can see the queries asked by other users.

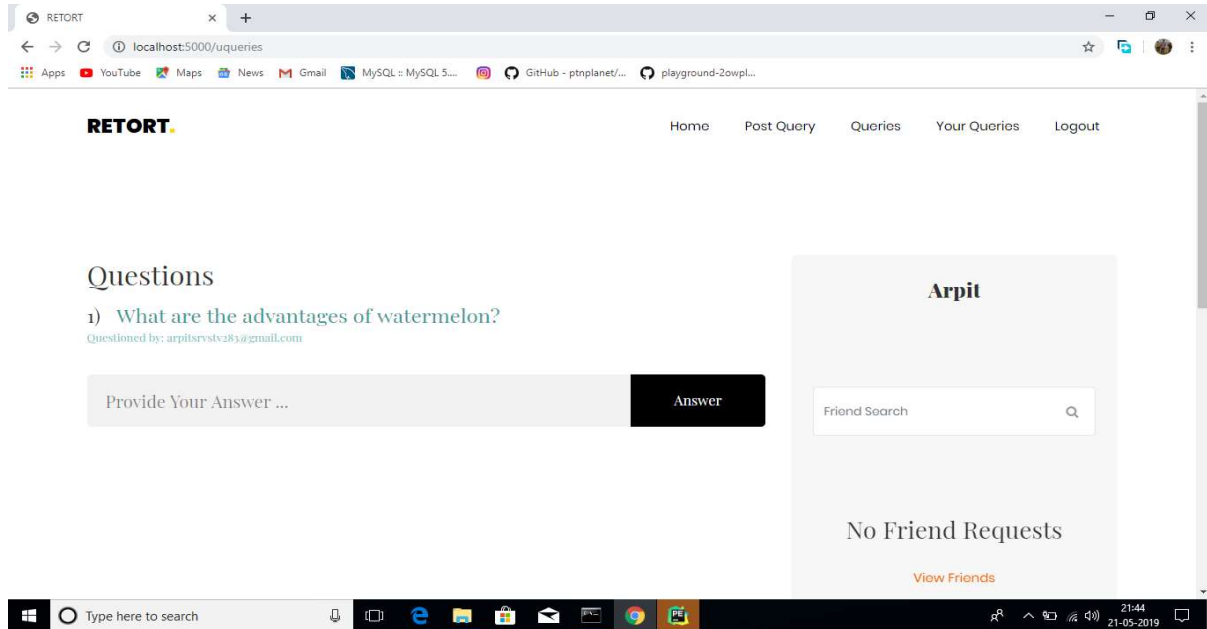


Figure 7.6: Other User Queries

In Figure 7.7, we can see our previous queries and their answers, which have been given by the user's friends.

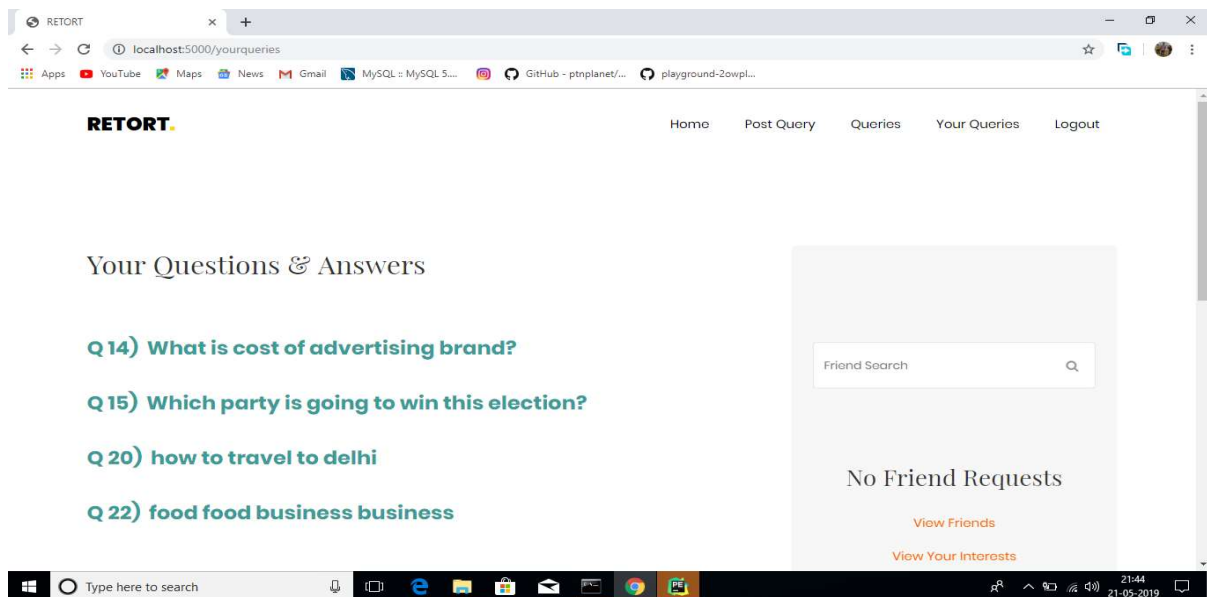


Figure 7.7: User Queries

RETORT

In Figure 7.8, we can see the admin dashboard after the admin has logged in.

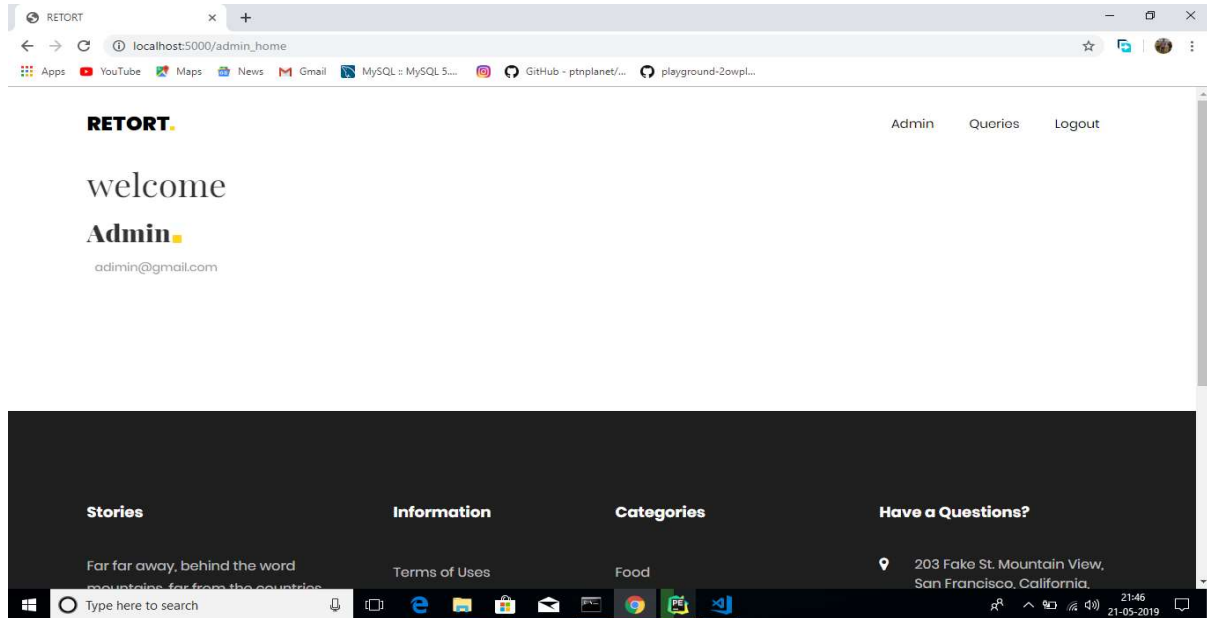


Figure 7.8: Admin Dashboard

In Figure 7.9, the admin processes each query, where we find the category of the question and it also decides to which friend the question is forwarded.

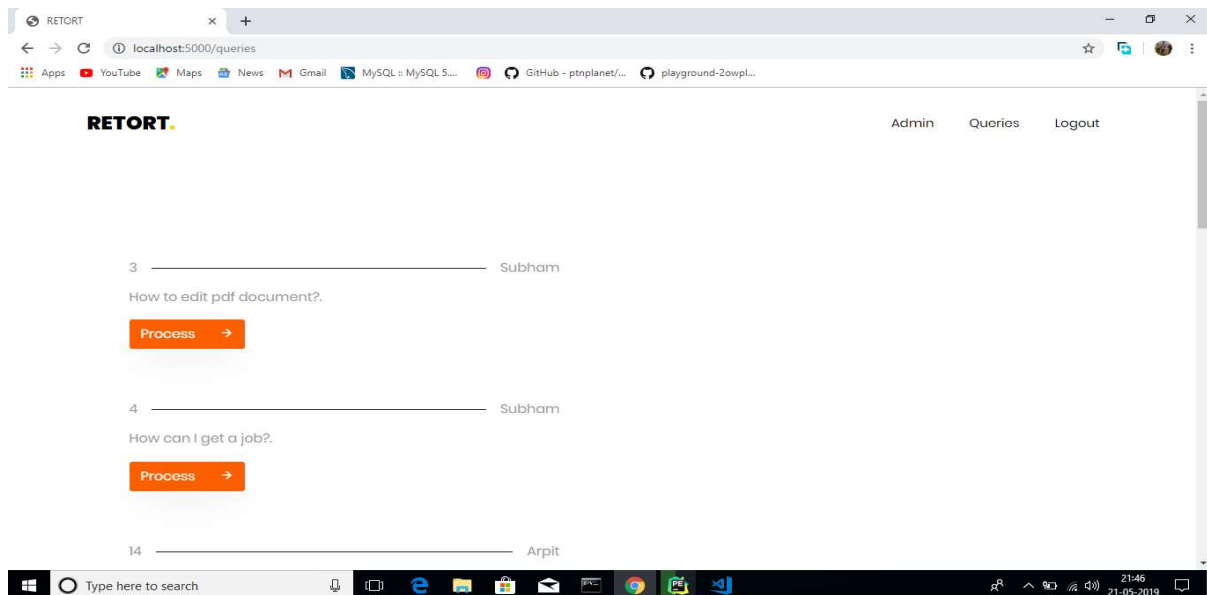


Figure 7.9: Admin Category

RETORT

In Figure 7.10, we see the questions asked by various other users.

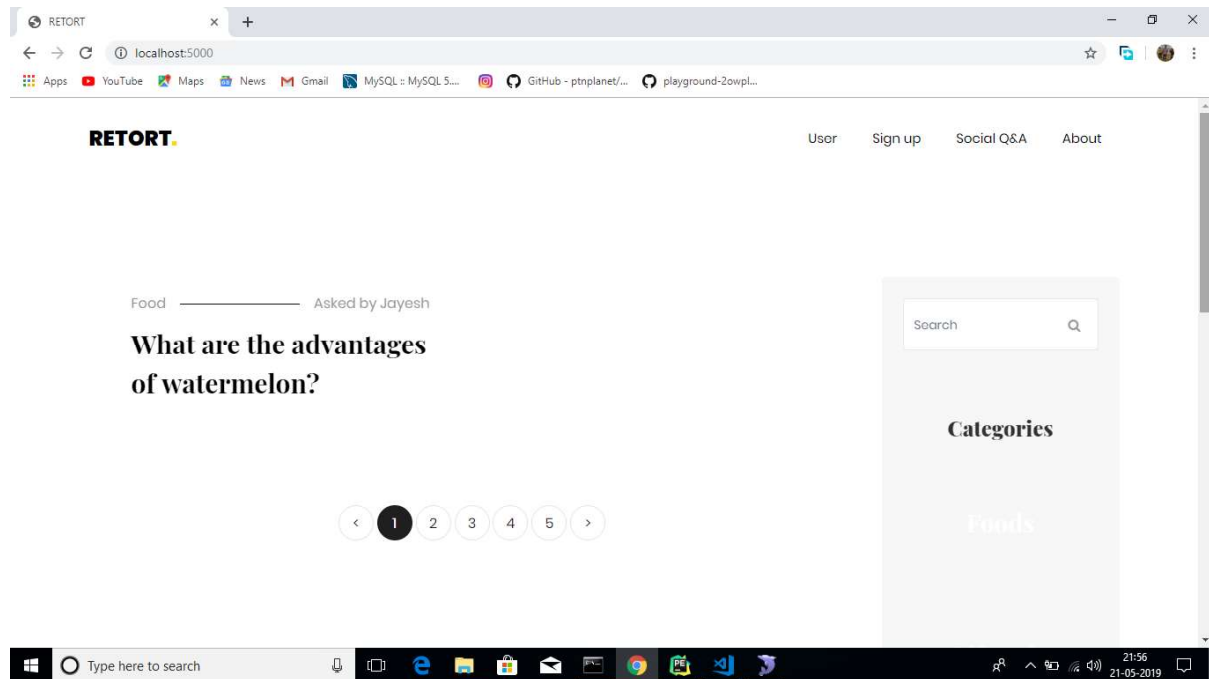


Figure 7.10: User Questions

Chapter 8

Conclusion and Future Scope

8.1 Conclusion

The RETORT is a network based Social Question and Answer platform, where different users can post their queries, view the queries which have been forwarded to them by different users and answer them. The user can also view also his/her other queries which he/she had asked previously along with their answers. The admin is subjected with the task of categorizing the question, along with forwarding it to the friend of that user.

8.2 Future Scope

The RETORT can be enhanced in the future with a wide range of different class labels with even a greater accuracy. An android application can be made for RETORT. While we are only using English, many other languages can also be provided. The query forwarding process can be automated by allowing it to directly classify the questions and forward it to the other users, without the intervention of the admin. The system can further be modified to answer the redundant question by itself, by searching the previous answers. Other techniques such as topic modelling and word embedding can be incorporated to find redundant question with a large scale user set.

References

- [1] Haiying Shen , Guoxin Liu , Haoyu Wang , Nikhil Vithlani. SocialQ&A: An Online Social Network Based Question and Answer System, 2017
- [2] G. Drosatos, P. Efraimidis, A. Arampatzis, G. Stamatelatos, and I. Athanasiadis. Pythia: A Privacy-enhanced Personalized Contextual Suggestion System for Tourism, 2015 .
- [3] Sabrina Gaito, Elena Pagani ,Gian Paolo Rossi. Contact Analysis in Workplace,2009 .
- [4] Matthew Richardson and Ryen W. White. Effects of expertise differences in synchronous social Q&A ,2012 .
- [5] M. R. Morris, J. Teevan, and K. Panovich. A Comparison of Information Seeking Using Search Engines and Social Networks. In In Proc. of ICWSM, 2010.
- [6] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge Sharing and Yahoo Answers: Everyone Knows Something. In Proc. of WWW, 2008.
- [7] A. Spagnolli and L. Gamberini. Interacting via sms: Practices of social closeness and reciprocation. British Journal of Social Psychology, 2007.
- [8] Ze Li, Haiying Shen Jin Li, Guoxin Liu. SOS: A Distributed Mobile Q&A System Based on Social Networks , 2013 .
- [9] B. Li, I. King, and M. R. Lyu. Question Routing in Community Question Answering: Putting Category in its Place. In Proc. Of CIKM, 2011.
- [10] L. Nie, Y. Zhao, X. Wang, J. Shen, and T. Chua. Learning to recommend descriptive tags for questions in social forums. TOIS, 2014.
- [11] N. A. Christakis and J. H. Fowler. Connected: The surprising power of our social networks and how they shape our lives. Hachette Digital, 2009.
- [12] A. Shtok, G. Dror, Y. Maarek, and I. Szpektor. Learning from the past: Answering new questions with past answers. In Proc. of WWW, 2012.