**RECOMMENDATION SYSTEM FOR**

**TOURISTS (TRIPSAHAYATRI)**

**A PROJECT REPORT**

****

**Submitted to**

**Department of Computer Science and Information Technology**

**Samriddhi College**

**In partial fulfillment of the requirements for the Bachelor’s Degree in Computer Science and Information Technology**

Submitted by

Popular Koju (5186/071)

Sandesh Shrestha (5202/071)

Subash Shrestha (5204/071)

Sumi Prajapati (5206/071)

OCTOBER 2018



**Supervisor’s Recommendation**

I hereby recommend that the project prepared under my supervision by **POPULAR KOJU, SANDESH SHRESTHA, SUBASH SHRESTHA AND SUMI PRAJAPATI** entitled "Recommendation System For Tourists (TripSahayatri)"be accepted as in fulfilling partial requirement for B.Sc. in Computer Science and Information Technology.

…………………………………….

**Mr. Sudip Khadka**

**SUPERVISOR**

**Faculty Member,**

**B.Sc. CSIT Department, Samriddhi College**

**Lokanthali, Bhaktapur**



**LETTER OF APPROVAL**

This is to certify that this project prepared by **Popular Koju, Sandesh Shrestha, Subash Shrestha and Sumi Prajapati** entitled "Recommendation System For Tourists (TripSahayatri)" in the partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

**Evaluation Committee**

|  |  |
| --- | --- |
| …………………………………….  **Mr. Sandeep Shrestha**  **Principal**  Samriddhi College | **…………………………………….**  **Mr. Sudip khadka**  **Supervisor,Faculty member**  Samriddhicollege |
| …………………………………….  **External Examiner**  Tribhuvan University  **Date**: | …………………………………….  **Internal Examiner**  Samriddhi College |

**DECLARATION**

We hereby declare that the project report entitled “**Recommendation system for tourists (Tripsahayatri)”** submitted to office of Dean, Faculty of Computer Scienceand Information Technology, Tribhuvan University, is our work done in the project work in the form of the degree of Bachelor Degree in Computer Science and Information Technology (B.Sc. CSIT) under supervision and guidance of **Supervisor Sudip Khadka**, Samriddhi College, Lokanthali, Bhaktapur, Nepal.

Popular Koju (5186/071)

Sandesh Shrestha (5202/071)

Subash Shrestha (5204/071)

Sumi Prajapati (5206/071)

# Acknowledgement

This project is the result of dedication and encouragement of many individuals. Our sincere and heartfelt appreciation goes to all of them. We would like to express my sincere thanks to our supervisor **Mr. Sudip Khadka**, for his support, motivation, suggestions and guidance. His advice was inevitable and with his help we were able to work on our own interested field and complete our project on time.

We are also thankful to **Mr. Sandeep Shrestha**, Principal, Samriddhi College who has provided all the help and facilities, which we required, for the completion of my project.

Moreover, we would like to express our heartfelt gratitude to all our teachers at Samriddhi College, Lokanthali, Bhaktapur who have imparted knowledge in various subjects.

Last but not the least; we would like to express our thanks to all our friends and our lovely parents for direct and indirect supports for the completion of this project.

# Abstract

Travelling becomes difficult when people don’t have clear vision about their destinations. Sometimes people do not have enough time to prepare themselves or they spend hours in some city without planning this before. Most of them ask friends or go for trips organized by tourist companies where a professional guide is involved. It will be very helpful if a system can recommend the places according to the user’s interest and the system that provides all information about the place to visit a city is available.

So regarding this we came up with this project that fulfills most of those needs. Our project entitled “Trip Sahayatri” is an android application that uses java programming language and is dedicated for the travelers. The system recommend them about the places they interest to visit. That be cultural sites, scenarios, religious places, trekking sites, etc. Collaborative filtering algorithm is used for the prediction and recommendation that analyze information about users with similar tastes to assess the probability so that a target individual will meet their choice. It uses Centered cosine similarity formula to filter out the most similar users. And from user reviews, preferences and ratings to make personalized recommendations to new places of that particular interested area.

Keywords: User-based Collaborative Filtering, Centered cosine, Recommendation

Table of Contents

[Acknowledgement IV](#_Toc526803007)

[Abstract V](#_Toc526803008)

[CHAPTER 1 INTRODUCTION 1](#_Toc526803009)

[1.1 Introduction 1](#_Toc526803010)

[1.2 Problem statement 2](#_Toc526803011)

[1.3 Objectives 3](#_Toc526803012)

[1.4 Scope of the system 4](#_Toc526803013)

[CHAPTER 2 REQUIREMENT ANALYSIS AND FEASIBILITY STUDY 5](#_Toc526803014)

[2.1 Literature Review 5](#_Toc526803015)

[2.2 Requirement Identification 7](#_Toc526803016)

[2.2.1 Functional Requirements 7](#_Toc526803017)

[2.2.2 Non-Functional Requirements 8](#_Toc526803018)

[2.3 Feasibility Study 9](#_Toc526803019)

[2.3.1 Economic Feasibility 9](#_Toc526803020)

[2.3.2 Technical feasibility 10](#_Toc526803021)

[2.3.3 Operational feasibility 10](#_Toc526803022)

[CHAPTER 3 SYSTEM DESIGN 11](#_Toc526803023)

[3.1 Process Design 11](#_Toc526803024)

[3.1.1 System Flow Diagram 11](#_Toc526803025)

[3.1.2 Sequence Diagram 12](#_Toc526803026)

[3.2 Database Design 13](#_Toc526803027)

[3.3 Class Diagram 14](#_Toc526803028)

[3.4 State Diagram 15](#_Toc526803029)

[CHAPTER 4 IMPLEMENTATION AND TESTING 16](#_Toc526803030)

[4.1 Implementation 16](#_Toc526803032)

[4.1.1 Implementation Tools 16](#_Toc526803033)

[4.2 Description of main modules 18](#_Toc526803034)

[4.3 Working of System 19](#_Toc526803035)

[4.4 Testing 21](#_Toc526803036)

[4.4.1 Unit Testing 21](#_Toc526803037)

[4.4.2 Integration Testing 24](#_Toc526803038)

[4.4.3 Result Analysis 25](#_Toc526803039)

[CHAPTER 5 CONCLUSION AND FUTURE ENHANCEMENT 26](#_Toc526803040)

[5.1 Conclusion 26](#_Toc526803042)

[5.2 Limitations 26](#_Toc526803043)

[References 27](#_Toc526803044)

[APPENDIX I 28](#_Toc526803045)

[Snapshots 28](#_Toc526803046)

[Source Code 30](#_Toc526803047)

[APPENDIX II 42](#_Toc526803048)

[Survey form used for data collection 42](#_Toc526803049)

[Database 43](#_Toc526803050)

**Table of Figures**

[**Figure 1: Use Case Diagram of the System** 7](#_Toc526803858)

[**Figure 2: Flow Diagram of the System** 11](#_Toc526803859)

[**Figure 3: Sequence Diagram of the System** 12](#_Toc526803860)

[**Figure 4: Class Diagram of the System** 14](#_Toc526803861)

[**Figure 5: State Diagram of the System** 15](#_Toc526803862)

**List of Tables**

[**Table 1: User Information** 13](#_Toc526803847)

[**Table 2: Place information** 13](#_Toc526803848)

[**Table 3: Rating** 13](#_Toc526803849)

[**Table 4: User authentication Testing** 21](#_Toc526803850)

[**Table 5: Firebase Authentication** 22](#_Toc526803851)

[**Table 6: Database Retrieval Error** 22](#_Toc526803852)

[**Table 7: Database Retrieval Solution** 22](#_Toc526803853)

[**Table 8: Data Structure Error** 23](#_Toc526803854)

[**Table 9: Data structure Solution** 23](#_Toc526803855)

[**Table 10: Integration Testing** 24](#_Toc526803856)

[**Table 11: Integration Testing Solution** 24](#_Toc526803857)

# CHAPTER 1 INTRODUCTION

# 1.1 Introduction

People travels a lot. People planning a trip often start investigations by looking for materials on the Internet. The user can find a large amount of information about the city and its popular tourism objects, but it takes a long time to select the most interesting places and create a good plan for a visit. There are lots of places to visit within the country and each places are different from each other on the basis of the culture, site, diversity and scenarios. It is quite difficult for anyone to choose the destinations according to their interest. They may have clear view about what they love but they may lack about the information of the places they are famous for and what kind of destination it is about. In this technological era everything we relate to our daily life is based on technology .In Tourism Industry also, for the visitors there is no proper system and resources for their guidance to make their desirable, easy and successful trip. Finding right destination is really a complex task in order to get the knowledge of the routes or information about the particular place they wish to visit. They don’t have prior knowledge about the places and what it is famous for.

So, our project is initiated with the purpose of recommending aid of information to peoples about the places they wants to travel on the basis of their interests. It can be cultural site, heritage site, trekking site, scenarios, and historical places and so on. They are given a sets of data’s to choose upon its interest and our system allows recommending attractions, which are better to attend and provide information on the basis of their preferences and past visits. Our system also contains popular places and helps them to get the best places. This will make a clear view to their destinations according to their interests and had a wonderful trip.

# 1.2 Problem statement

Nowadays, people aims to travel a lot. They don’t have idea about the initiation of the plans. Sometimes, their desire of travelling, just lost somewhere in their dreams only because they don’t have access to any guidance for that particular place and information. They might not be familiar about the places they want.When they don’t have proper vision about these things then they might not be able to visit the places of their interest. We can get maps and obtain information about museums, galleries, tourist attractions, and probably entrance costs. Most of them we can find on the web page designed for this place, but often these all the attractions may not be the choice of interest for all the visitors and searching on the internet becomes more confusing and tedious task to perform. So we thought of a system that easily provides all the information. The information about the places is based on the visitor’s interest .It then recommends them appropriate destination where they can visit on their basis of their interest and ratings. Our system helps tourist’s to turn their dream destinations into reality.

# 1.3 Objectives

The proposed system has the following objectives:

* To recommend the places to visit for the tourists on the basis of their interests.
* To provide information about that place.
* To suggest the destination on the basis of similar users ratings and preferences.
* To list out the popular and top related places to visitors.

# 1.4 Scope of the system

Many of us nowadays rely on the new technologies and applications and we want everything to be easy and on perfect way. Our application can play important role to the tourists for their better services. It will facilitate the travelers who loves travelling and visiting different places but don’t have proper view and knowledge about the places and don’t have access to any guidance for that particular place. Through our application tourists can access accurate information about the kind of place easily. It will make them access about the place whenever required their by improving its operational efficiency and effectiveness.

# CHAPTER 2 REQUIREMENT ANALYSIS AND FEASIBILITY STUDY

## 2.1 Literature Review

To identify various requirements for our android mobile based application we have studied various papers of existing system and reviewed different applications that followed recommendation algorithms and system such as:

* Tourist Guide System [1]

This study of system was published on Imperal Journal of Interdisciplinary Research (IJIR), vol-2, issue-3, 2016.Issn:2454-1362.This research was conducted by Professor S.S.Parwar, Pooja Chvan and Arty Lohar in Scoe Vadgaon, Pune, India. An android system which helps tourists to find the better place at one instant. It provides information about hotels, restaurants and museum at one place and helps the need of the tourists in very efficient, quicker and flexible manner.

* Tourist attraction information service [2]

This project was proposed by, Maksim Achekotov institute for informatics and automation of the Russian academy of sciences St. Petersburg, Russia. This paper proposed a description of information decision support system in the tourist domain and set of methods and algorithms for generating recommendations for a user that allowed significant increase of the system usability. This system allowed recommending attractions, which are better to attend based on tourist preferences and the context information of the location area.

* Taxidio

Taxidio [3] is an android application developed by Travel India private Ltd. 302, Doli Chambers in Mumbai released on 26 march 2018.It is an automated trip planner that understands your unique travel preference and recommends the ideal holiday’s destination for you, enables you to create fully customizable itineraries, helps you book your accommodation and attraction tickets, all under one roof.

The above system were studied and reviewed to grab the User interface and functionings of the system for the application development and UI.

Mainly,the proposed system developed aims to recommend the places based on users interests and recommend places to them providing detail information.

On performing research the paper by Patcharee, Srisuwan and Anongnart Srivihok published in Bangkok was found, that described about personalized trip information for E-tourism. This paper presents the personalized recommendation system for e-tourism by using statistic technique base on Bayes Theorem to analyze user behaviour and recommend trips to specific users. The system is evaluated by using Recall, Precision and F-measure. Results demonstrate that it is possible to develop Personalization Recommendation System. Past and recent information of customer's behaviors are used for recommending products and services best fit to relevant customers in the e-Tourism.

Likewise, Wahidah Husain and Lam Yih Dih presented a Framework of a Personalized Location-based Traveler Recommendation System [4] in Mobile Application.The goal of this research was to propose a suitable recommendation method for use in a Personalized Location-based Traveler Recommender System (PLTRS) to provide personalized tourism information to its users. A comparative study of available recommender systems and location-based services (LBS) was conducted to explore the different approaches to recommender systems and LBS technology.

So, through all this research knowing all the drawbacks and feedbacks of available recommendation systems we had extracted and implemented personalized User-based [5] recommendation algorithm in our System. It works by comparing and considering the feature of active user and original user who would have the similarity user database. The results are based on the feedback from users who are similar to the target user instead of on the target user's own past preferences. The accuracy of a collaborative filtering [6] method depends on the number of items which can be associated with certain users. In addition, new items or information added to the database are neglected by the recommender system as there is no initial rating for the item. These scalability, accuracy and “cold start” [7] problems of a collaborative filtering recommender system are issues to be concerned about, especially in a mobile application.

## 2.2 Requirement Identification

Requirements are the description of the services that a software system must provide and the constraints which it operates. Requirements can range from high-level abstract statements of service or system constraints to detailed mathematical functional specifications. There are two types of requirements:

Functional Requirements and Non-Functional Requirements

### 2.2.1 Functional Requirements

Functional requirements include the functions performed by specific screens, outlines of work-flows performed by the system and other requirements that the system must meet.

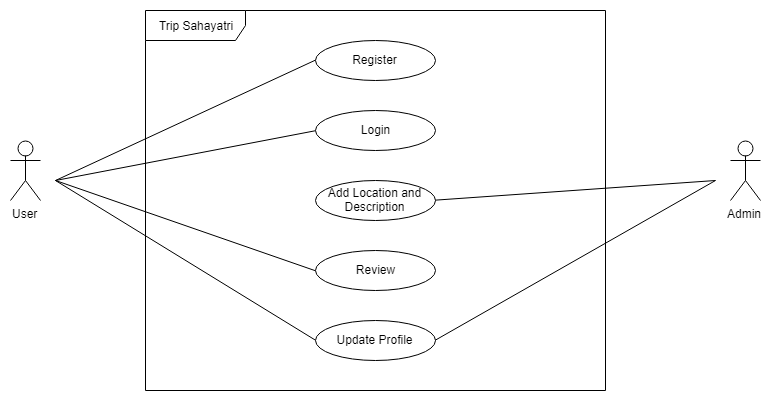
Developed system should have following functional requirements.

* System should recommend place according to the interest of tourists.
* System should provide information of recommended place.
* Travelers should be able to rate tourism places after visiting.
* Traveler should get top popular similar places to visit.

The Use Case diagram [8] of functional requirement is given below:

Use case diagram is a graphical depiction of the interaction among the element of the system. A use case diagram main components:

* The Boundary: defines the system of interest
* The Action: usually individual involved with the system defined according to the role.
* The Use case: specific role are played by action within and round the system.
* The Relationship between and among the action and the use case.



**Figure 1: Use Case Diagram of the System**

### 2.2.2 Non-Functional Requirements

Non-functional requirements cover all the remaining requirements which are not covered by functional requirements. The specific criteria that judge the operation of a system, rather than specific behaviors. The non-functional requirement of the system are:

* Security:

Security requirement are needed in system to ensure unauthorized access to system and its data is not allowed as well as to ensure the integrity of the system from accidental or malicious damage.

* Reliability:

Reliability is the ability of a system to perform its required function under stated condition for a specific period of time.

* Performance:

It concerns with the speed of operation of a system like giving the possible accurate results.

* Maintainability:

This is the ability to change the system to deal with new technology or to fix the defects.

* Usability:

Usability is the ease with which user can learn to operate, prepare inputs for and interpret outputs of system or components. User-manuals, Help facilities are some of the components that are included in usability

## 2.3 Feasibility Study

Feasibility analysis is carried out in order to check whether our project is feasible enough to operate in various places or not. Our system must be feasible so that it can be implemented in time and used easily in various browsers. The main feature required for project to be a feasible are of mainly four types which are as follows:

### 2.3.1 Economic Feasibility

This study is undertaken in order to analyze the benefit that we achieved from the cost incurred from the project or system is referred to as cost benefit analysis. This is vital to understand how feasible the project is economically.

The total cost of the project includes both the monetary and non-monetary cost. Monetary costs are incurred upon implementation and throughout the life of the project. These include start-up fees for Wi-Fi, power supply, travel, training, testing and production materials, payroll expenses, user acceptance processes, training, and travel expenses, among others. The non-monetary cost  include time, lost production on other tasks, imperfect processes, potential risks, market saturation or penetration uncertainties, and influences on one’s reputation.

### 2.3.2 Technical feasibility

Technical resource must be upgraded or added to in manner that fulfills the desire and request under consideration. And every type of technology is required in order to undertake the project and to complete it on time. The main aim of technical feasibility is to support the cost of company to undertake a technical study into development of new products, technologies, processes or internationally trade services. Technical feasibility depends upon the existing computer hardware and to what extent it can support the proposed addition. This involves financial consideration to accommodate technical enhancements. If budget is serious constraint then it is not taken as feasible project. So it measure whether project is technically feasible or not.

The system is technically feasible if the proposed technology is easily available to the clients at fewer expenses.

### 2.3.3 Operational feasibility

The project is operationally feasible as the android device required for running the application is easily accessible and AVD (Android Virtual Device) supporting machine is used for the development which include all the Google API support.

# 

# CHAPTER 3 SYSTEM DESIGN

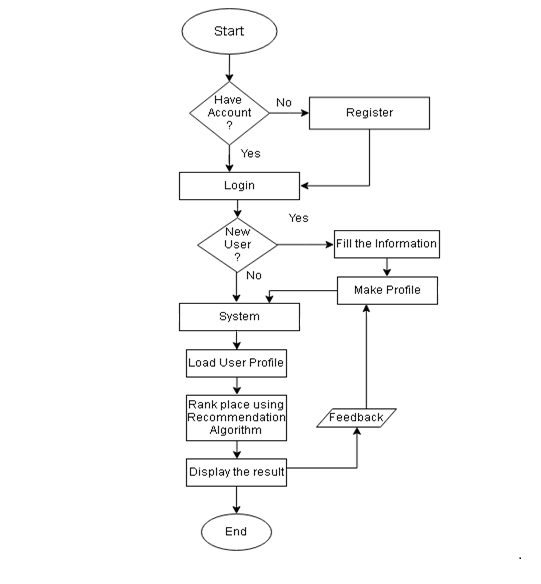
## 3.1 Process Design

Process design is the activity of determining the workflow, equipment needs and implementations requirements for a particular process. Process design typically uses a number of tools including flowcharts, process simulation software and scale methods.

### 3.1.1 System Flow Diagram

A flowchart is a formalized graphic representation of a logic sequence, work or organizational chart, or similar formalized structure. The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process.

Flowcharts use simple geometric symbols and arrows to define relationships. In programming, for instance, the beginning or end of a program is represented by an oval. A process is represented by a rectangle, a decision is represented by a diamond and I/O process is represented by a parallelogram. The following flowchart depicts the flow of our program.



**Figure 2: Flow Diagram of the System**

### 3.1.2 Sequence Diagram

A sequence diagram [8] is an interactive diagram that shows how object operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. Sequence diagram are typically associated with use case realization in the logical view of the system under development. Sequence diagram are sometimes called event diagram or event scenarios. The sequence diagram for this system is shown below:

****

**Figure 3: Sequence Diagram of the System**

## 3.2 Database Design

The database schema is a specification that describes the structure of the database such as

tables and indexes, identifies the stored procedures that access data in the database, and

defines the way tables and stored procedures are partitioned for fast data access. When

designing client applications to use the database, the schema specifies the details needed

about data types, tables, columns, and so on.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraint | Key |
| id | varchar (255) | NOT\_NULL | Primary\_key |
| name | varchar (255) | NOT\_NULL |  |
| username | varchar (255) | NOT\_NULL |  |
| email | varchar (255) | NOT\_NULL |  |
| password | varchar (255) | NOT\_NULL |  |
| interest | varchar (255) | NOT\_NULL |  |

**Table 1: User Information**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraint | Key |
| pid | varchar (255) | NOT\_NULL | Primary\_key |
| name | varchar (255) | NOT\_NULL |  |
| address | varchar (255) | NOT\_NULL |  |
| description | text | NOT\_NULL |  |
| lng | varchar (255) | NOT\_NULL |  |
| ltd | varchar (255) | NOT\_NULL |  |
| type | varchar (255) | NOT\_NULL |  |
| url | text | NOT\_NULL |  |

**Table 2: Place information**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Datatype | Constraint | Key |
| uid | varchar (255) | NOT\_NULL | FOREIGN |
| pid | varchar (255) | NOT\_NULL | FOREIGN |
| rating | varchar (255) | NOT\_NULL |  |

**Table 3: Rating**

## 3.3 Class Diagram

There are five main classes used in this system:

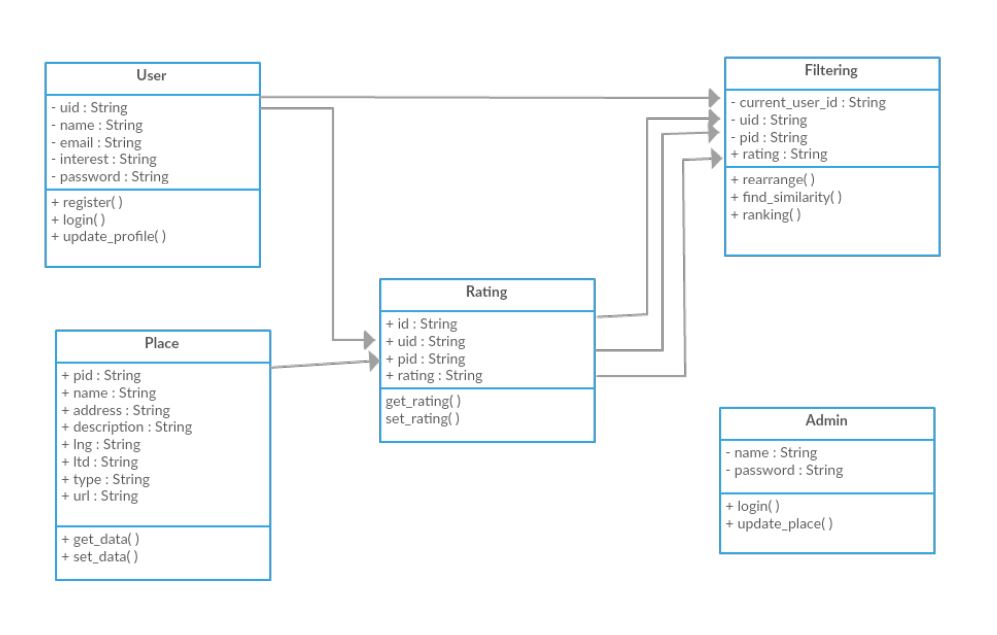
It contains a list of attributes i.e. uid, name, username and password of each user. The user needs to enter the details while registering to the application. The user class consists of methods like register(), login() and update\_profile() for the input in the system.

The place class consist of attributes like pid, name, address and description of each place. It also contains the longitude, latitude of that place with its url for the description of that place. The get\_data() method is used to get the data from the database and set\_data() method updates the new places in the database.

The rating class consist of attributes i.e uid, pid and rating provided by different user to different places. There are two methods get\_ratings and set\_ratings that are used for retrieving the rated places from the user and set that ratings in the database.

The Filtering class consists of attributes i.e. uid, pid, ratings and current\_user\_id. The methods in this class are used for the rearrangement of the data after filtering and find the similarity between the current user and previous similar users. The ranking is done on the basis of that ratings.

Admin class is used for login and only updating the places in the database. It doesn’t involves in different methods in the system of other classes.



**Figure 4: Class Diagram of the System**

## 3.4 State Diagram

This figure explains the state of application.



**Figure 5: State Diagram of the System**

# CHAPTER 4 IMPLEMENTATION AND TESTING



## 4.1 Implementation

### 4.1.1 Implementation Tools

The “Trip Sahayatri” application is developed by constructing a customized User Interface by targeting to provide simple and user-friendly interface to the users. The user-friendly interface is maintained so that users can use this application without any difficulties.

The whole information gathered is studied, analyzed and then finally it was processed to build an actual system.

Different implementation tools were used in this system such as:

#### 4.1.1.1 Java

The development of android application has been done using Java language which is an object-oriented programming language.

#### 4.1.1.2 XML

For the layout definition of the application and training data sets, XML has been used explicitly.

#### 4.1.1.3 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for android app development, based on IntelliJ IDEA .On top of IntelliJ’s powerful code editor and development tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

* A flexible Gradle-based build system
* Instant Run to push changes to running app without building a new APK
* Code templates and GitHub integration help to build common app features and import sample code
* Extensive testing tools and framework
* Lint tools to catch performance ,usability, version compatibility, and other problems
* C++ and NDK support

#### 4.1.1.4 Git and GitHub

GIT is a very popular source code version control system (VCS).It is distributed revision control system with an emphasis on speed, data integrity, and support for distribution, nonlinear workflows. Ever GIT working directory is a full-fledged repository with complete history and full version tracking capabilities, independent of network access and central server.

GitHub is a web-based hosting service for projects. It is distributed version control that makes it easy to collaborate. Some features of GitHub are:

* Review code efficiently with pull requests and commit histories
* Hold discussions right in the source code with inline comments
* Filter the view by branch or tag to see team progress
* GIT is used to maintain the code in an online GitHub repository for the project

#### 4.1.1.5 Microsoft Office Word

Microsoft Office Word 2013 is used in this project for the documentation of the final year project.

#### 4.1.1.6 Creately

Creately is used in this project to draw data flow diagram, use case diagram and Gantt chart which provides the necessary shapes.

## 4.2 Description of main modules

In this whole process, there is presence of a large number of modules, which are listed below:

1. **PHP API for data storage and retrieval**

For the storage and retrieval of data from the database we have used PHP programming language and MySQL and then generate API to be used in android application since Android Studio doesn’t support MySQL database. This API is mainly used to take data from Android app and store it into the database as well as retrieved the data from database to display in Android application. PHP API generates JSON file and it is parsed using volley in Android.

1. **Data rearrangement module**

This module takes the data from rating table of database and rearrange into the form from which it will be easy for implementing in other module such as calculating similarity and ranking of places. For this the system used data structure knowledge such as Treeset, HashMap and Map.

1. **Similarity module**

We used user based collaborative filtering for recommendation. There are various different recommendation algorithms in it. Among all of them the system used centered cosine similarity method to extract the nearest and most similar user as it was effective and correct.This modules takes rearranged data of rating table and compare it with current user profile in order to find out most similar user.

1. **Ranking module**

From the similarity value or the user’s similarity we rank the place available in the system which are not visited by current user but are visited by most similar user. We used min max algorithm to rank the place on the basis of high rating.

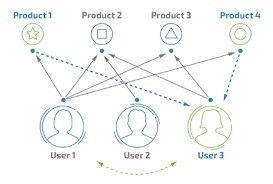
1. **Result**

From the above similarity module and ranking module, the recommended places were observed. They were according to user’s interest and on the basis of previous user’s rating. They were accurate and system was able to make a close recommendation to user’s interest.

## 4.3 Working of System

**User based Collaborative filtering**

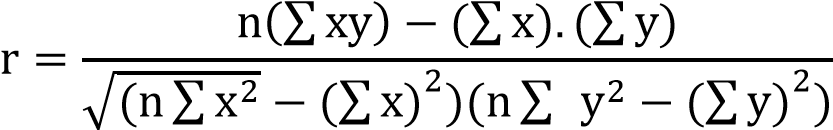
This filtering methodology is used to recommend items to the user based on the rating of other users having similar preferences. The algorithm works perfectly fine when the number of users and items are less. But, when the number of items starts increasing problems of data sparsity occurs.



**Algorithm**

1. Collect preference of all the users listing the ratings of users on different places.

2. Calculate similarity between the users using Pearson Correlation Coefficient.



where,

n = number of pairs of scores

Σxy = sum of products of paired scores

Σx = sum of x scores

Σy = sum of y scores

Σx2 = sum of squared of x scores

Σy2 = sum of squared of y scores

3. Sort the similarity scores between the users such that the users in descending order.

4. Produce weighted score that ranks the users by taking the multiplying their ratings of different places with the similarity score.

**Pseudocode**

u = currentUser()

S\_u = usersSimilarTo(u)

for i in itemsRatedBy(S\_u) and i not in itemsRatedBy(u):

s = score(i, u, S\_u)

recommendations += (i, s)

sortByScore(recommendations)

return recommendations

## 4.4 Testing

Testing is the process of evaluating a system or its components with an intent to find whether it satisfies the expected requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Various types of testing procedures were performed in order to check the working mechanism and correctness of the system. Some of the types of testing that we did attempt are described below:

### 4.4.1 Unit Testing

The purpose of unit testing in this project is to check that as individual parts of the system are functioning as expected or not. We have tested each module of this system in order to check the correctness of the output. Firstly, we found many errors and then it was resolved stepwise.

**Test Case 1**

Objective of test case 1 is to check whether the input for login is correct or not and user does exist before. Input given is the username and password. Expected output is to ensure authenticated email address and password.

|  |  |
| --- | --- |
| Test Name | Unit Test |
| Objective | If the user wants to login inserted username and password must be valid |
| Input | Username, Password |
| Output | Login unsuccessful |
| Error Information | User doesn’t exist |

**Table 4: User authentication Testing**

Solution:

There was problem and user weren’t registered or username exists of same name.

|  |  |
| --- | --- |
| Test Name | Unit Test |
| Objective | If user wants to login, inserted username should have been registered and must be a valid username and password. |
| Input | Username, Password |
| Expected output | Login successful |
| Method | Firebase authentication |
| Original output | Successfully login |

**Table 5: Firebase Authentication**

**Test Case 2**

Objective of test case 2 is to check whether the data is retrieved from the stored database using asynchronous method.

|  |  |
| --- | --- |
| Test Name | Unit Test |
| Input | Data from Android UI |
| Expected output | Data retrieved from Android UI to PHP API |
| Output | Error and data not retrieved |
| Error information | Implementation error |

**Table 6: Database Retrieval Error**

Solution:

As data retrieval was unsuccessful we used MySQL, Android and JSON to retrieve the data and PHP API was made.

|  |  |
| --- | --- |
| Test Name | Unit Test |
| Input | Data from Android UI to PHP API. |
| Output | Result obtained and successfully retrieved |

**Table 7: Database Retrieval Solution**

**Test Case 3**

Objective of test case 3 is to check whether the data is rearranged in structure format.

|  |  |
| --- | --- |
| Test Name | Unit Test |
| Input | Ratings from database |
| Expected output | Users ratings in restructured format |
| Output | Unstructured and random output |

**Table 8: Data Structure Error**

Solution:

As user’s ratings is obtained random we used data structured knowledge such as HashMap, Map and Treeset.

|  |  |
| --- | --- |
| Test Name | Unit Test |
| Input | User’s rating using HashMap, Map and Treeset |
| Output | Structured format in rearranged data form |

**Table 9: Data structure Solution**

### 4.4.2 Integration Testing

The purpose of integration test is to verify the functional, performance and reliability between the modules that are integrated.

**Test Case 4**

Objective of test case 4 is to check whether the recommended place is according to user’s interests or not.

|  |  |
| --- | --- |
| Test Name | Integration Testing |
| Input | User profile structured rating |
| Method | Cosine similarity |
| Expected output | Place according to user’s interest |
| Output | Accurate interested place was not displayed |
| Error info | No user based collaborative filtering and ratings were not actual. |

**Table 10: Integration Testing**

Solution:

As place according to user’s interest were not obtained, Pearson correlation/Centered cosine similarity is used.

|  |  |
| --- | --- |
| Test Name | Integration Test |
| Input | User profile, structured rating |
| Method | Pearson correlation/Centered cosine |
| Expected output | Recommended place according to user’s input interest from user profile |
| Output | Accurate and recommended place is according to user’s interest |

**Table 11: Integration Testing Solution**

### 4.4.3 Result Analysis

After many days of research, analysis and implication of knowledge, we finally built an Android mobile app. The component of Android app is functioning properly as expected and we had developed beta version. This application can be used by user from anywhere.

“Tripshayatri” is an android based application which is still under development and even the functioning parts are in various pieces. This android application is designed from the user point of view. The user friendly design enables the user to accomplish their requirement with the ease.

Overall, this application is very beneficial for general users having the Android phones which aim to travel different places on their choices.

# CHAPTER 5 CONCLUSION AND FUTURE ENHANCEMENT



## 5.1 Conclusion

The main interaction of this project is to provide an application interface that simplifies travelling experience for peoples. Knowing the personal interest and recommending on the basis of users reviews the application suggests the places for users and guide them to travel. While developing such program, we came across various mobile app development techniques and have gained knowledge on various aspect of programming such as mobile app development, Java, XML and so on.

## 5.2 Limitations

There are some limitations in our application which will be solved in our future versions. Some of the limitations are listed below:

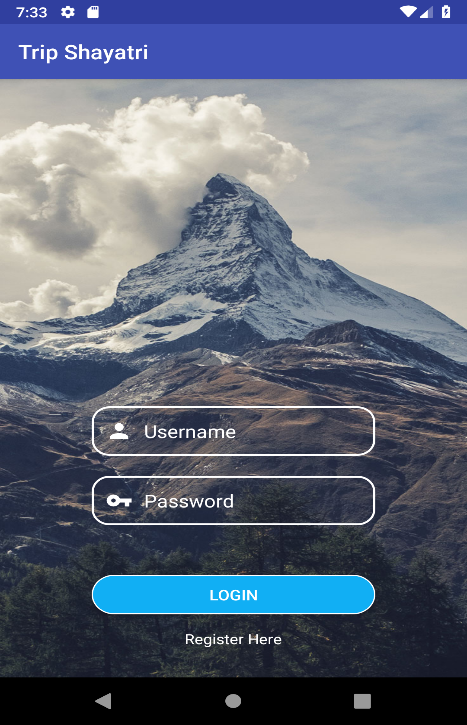
* This application runs only on Android platform i.e it requires devices with Android OS version of 5.0 or higher.
* This system contains only the limited number of data sets.
* This system may not give more accurate results and the basic information and solutions may not meet the user needs.
* This system does not recommend hotels, lodges, weather condition of the places to the users.

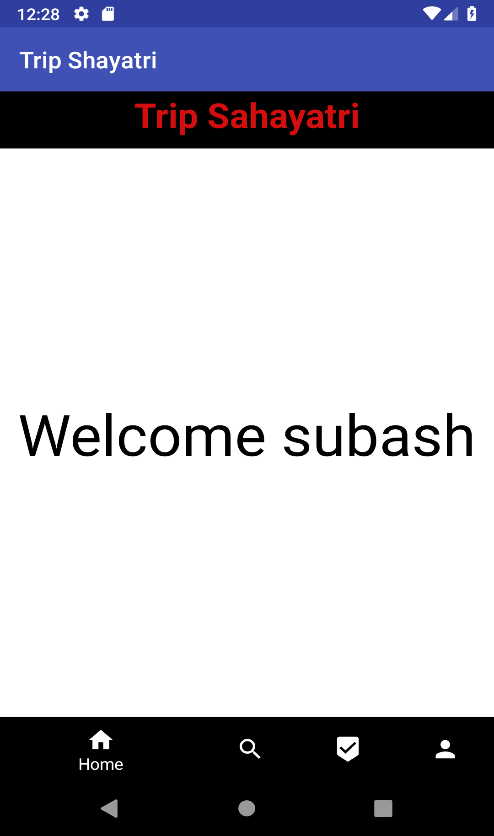
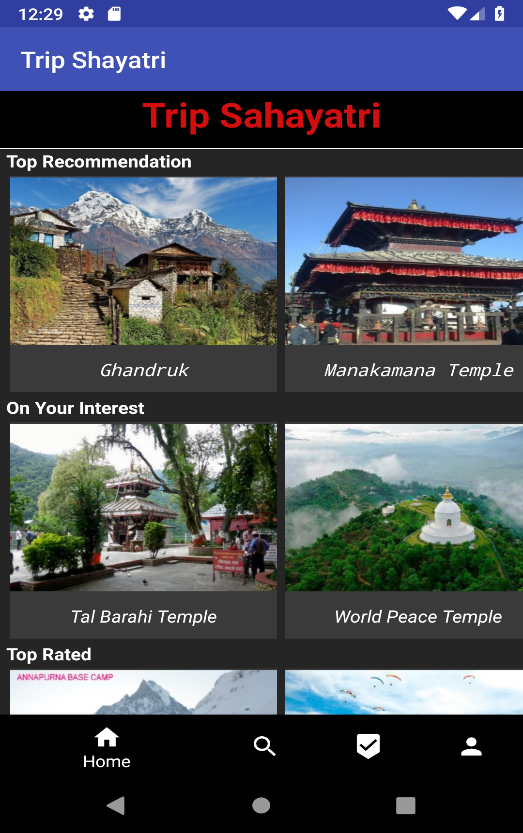
# References

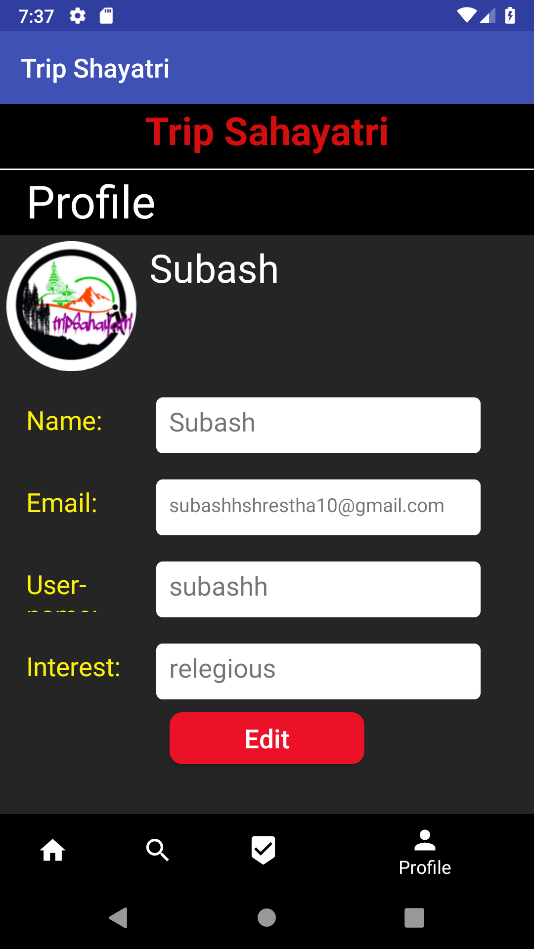
|  |  |
| --- | --- |
| [1] | P. C. a. A. L. S.S.Parwar, "Tourist Guid System," *Imperal Journal of Interdisciplinary Research(IJIR),* vol. 2, no. 3, 2016. |
| [2] | A. K. A. P. N. S. Alexander Smirnov, "Tourist Attraction System," Makism Achekotov institute if information and automation, Russia. |
| [3] | T. I. P. Ltd, "Taxido," Mumbai, 2018. |
| [4] | W. H. a. L. Y. Dih, "Location Based Traveler Recommendation System," Bangladesh, 2014. |
| [5] | Z.-D. Z. a. M.-s. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," in *Third International Conference on Knowledge Discovery and Data Mining*, Thiland, 2010. |
| [6] | D. H. a. C. K. John S. Breese, "Empirical analysis of predictive algorithms for collaborative filtering," *Technical Report MSR-TR-98-12,* 1998. |
| [7] | T. V. D. L. a. A. D. D. Xuan Nhat Lam, "Addressing cold-start problem in recommendation system," *ACM Digital Library,* pp. 208-2011, January 31 - February 01, 2008 . |
| [8] | J. F. a. J. S. Jeffrey A. Hoffer, Modern Systems Analysis and Design, Delhi: Pearson, 2014. |

# APPENDIX I

## Snapshots





## Source Code

Homepage.java

public class HomePage extends AppCompatActivity {

BottomNavigationView bottomNavigationView;

FrameLayout mainFrame;

HomeFragment homeFragment;

ProfileFragment profileFragment;

SearchFragment searchFragment;

VisitedFragment visitedFragment;

TextView textView;

String currentuserid,name;

RequestQueue requestQueue;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.homelayout);

textView=findViewById(R.id.welcomview);

mainFrame=findViewById(R.id.mainframe);

bottomNavigationView =findViewById(R.id.navigation);

homeFragment=new HomeFragment();

profileFragment=new ProfileFragment();

searchFragment=new SearchFragment();

visitedFragment=new VisitedFragment();

currentuserid= FirebaseAuth.getInstance().getCurrentUser().getUid();

requestQueue = Volley.newRequestQueue(getApplicationContext());

getUser(currentuserid);

bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item) {

switch (item.getItemId()){

case R.id.home:

setFragment(homeFragment);

return true;

case R.id.search:

setFragment(searchFragment);

return true;

case R.id.visited:

setFragment(visitedFragment);

return true;

case R.id.profile:

setFragment(profileFragment);

return true;

default:

return false;

}

}

});

}

private void setFragment(Fragment fragment) {

FragmentTransaction fragmentTransaction=getSupportFragmentManager().beginTransaction();

fragmentTransaction.replace(R.id.mainframe,fragment);

fragmentTransaction.commit();

}

public void getUser(final String id){

StringRequest stringRequest = new StringRequest(Request.Method.POST, "http://192.168.137.1/subash/getuserinformation.php", new Response.Listener<String>() {

@Override

public void onResponse(String response) {

try {

JSONObject obj1 = new JSONObject(response);

JSONArray a1=obj1.getJSONArray("user");

JSONObject obj2=a1.getJSONObject(0);

name=obj2.getString("name");

textView.setText("Welcome "+name);

}

catch(Exception e){

Log.d("VAL", ""+e);

Toast.makeText(getApplicationContext(), "Internal Faliure", Toast.LENGTH\_LONG).show();

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

// progressDialog.dismiss();

Log.d("VAL", ""+error);

Toast.makeText(getApplicationContext(), "Error in connection.", Toast.LENGTH\_LONG).show();

Log.d("VAL", ""+error);

}

}){

@Override

protected Map<String, String> getParams() throws AuthFailureError {

Map<String, String> myMap = new HashMap<>();

myMap.put("id", currentuserid);

return myMap;

}

};

requestQueue.add(stringRequest);

}

}

MainActivity.java

Packagecom.example.de.tripshayatri;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity {

EditText email,password;

Button login;

FirebaseAuth auth;

TextView register;

ProgressDialog dialog;

@Override

protected void onCreate(Bundle savedInstanceState) {

auth=FirebaseAuth.getInstance();

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

email=findViewById(R.id.username);

password=findViewById(R.id.password);

dialog=new ProgressDialog(MainActivity.this);

dialog.setMessage("Loading");

dialog.setCancelable(false);

login=findViewById(R.id.login);

register=findViewById(R.id.register);

login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String myemail=email.getText().toString();

String mypassword=password.getText().toString();

if (TextUtils.isEmpty(myemail) || TextUtils.isEmpty(mypassword)) {

Toast.makeText(MainActivity.this, "Empty Username or Password", Toast.LENGTH\_SHORT).show();

}

else{

dialog.show();

auth.signInWithEmailAndPassword(myemail,mypassword).addOnSuccessListener(new OnSuccessListener<AuthResult>() {

@Override

public void onSuccess(AuthResult authResult) {

// SharedPreferences sp1 = getSharedPreferences("yourfile", Context.MODE\_PRIVATE);//for saving login session

// SharedPreferences.Editor editor = sp1.edit();//shared preferance lai edit garna lai editor banako

// editor.putBoolean("state", true);//login state lai true gareko

// editor.commit();

dialog.dismiss();

Toast.makeText(MainActivity.this, "Login Sucessfull", Toast.LENGTH\_SHORT).show();

Intent i = new Intent(MainActivity.this, HomePage.class);

startActivity(i);

//data from mysql

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

dialog.dismiss();

Toast.makeText(MainActivity.this, "Login Faliure", Toast.LENGTH\_SHORT).show();

}

});

}

//ya logic lekhna baki cha

}

});

register.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i= new Intent(MainActivity.this,Register.class);

startActivity(i);

}

});

}

}

DashActivity.java

package com.example.de.tripshayatri;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

import com.android.volley.AuthFailureError;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;

import com.android.volley.toolbox.Volley;

import com.google.firebase.auth.FirebaseAuth;

import org.json.JSONArray;

import org.json.JSONObject;

import java.util.HashMap;

import java.util.Map;

public class DashActivity extends AppCompatActivity {

TextView t1;

String currentuserid;

String[] uname=new String[100];

String id,name,username;

RequestQueue requestQueue;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.dashlayout);

currentuserid= FirebaseAuth.getInstance().getCurrentUser().getUid();

uname=getUser(currentuserid);

t1=findViewById(R.id.view);

requestQueue = Volley.newRequestQueue(getApplicationContext());

uname=getUser(id);

t1.setText("Welcome "+uname[0]);

t1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

t1.setText("Welcome "+uname[0]+"Your name is "+uname[1]);

}

});

}

// from here fetching data from database

public String[] getUser(final String id){

final String[] uname=new String[100];

StringRequest stringRequest = new StringRequest(Request.Method.POST, "http://192.168.137.1/subash/getuserinformation.php", new Response.Listener<String>() {

@Override

public void onResponse(String response) {

try {

JSONObject obj1 = new JSONObject(response);

JSONArray a1=obj1.getJSONArray("user");

JSONObject obj2=a1.getJSONObject(0);

name=obj2.getString("name");

username=obj2.getString("username");

uname[0] =username;

uname[1]=name;

// t1.setText("Welcome "+uname[0]);// ya bata set garda chai display vayo

Toast.makeText(DashActivity.this, username, Toast.LENGTH\_SHORT).show();

}

catch(Exception e){

Log.d("VAL", ""+e);

Toast.makeText(DashActivity.this, "Internal Faliure", Toast.LENGTH\_LONG).show();

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

// progressDialog.dismiss();

Log.d("VAL", ""+error);

Toast.makeText(DashActivity.this, "Error in connection.", Toast.LENGTH\_LONG).show();

Log.d("VAL", ""+error);

}

}){

@Override

protected Map<String, String> getParams() throws AuthFailureError {

Map<String, String> myMap = new HashMap<>();

myMap.put("id", id);

return myMap;

}

};

requestQueue.add(stringRequest);

return uname;//returning array of all the user

}

}

# APPENDIX II

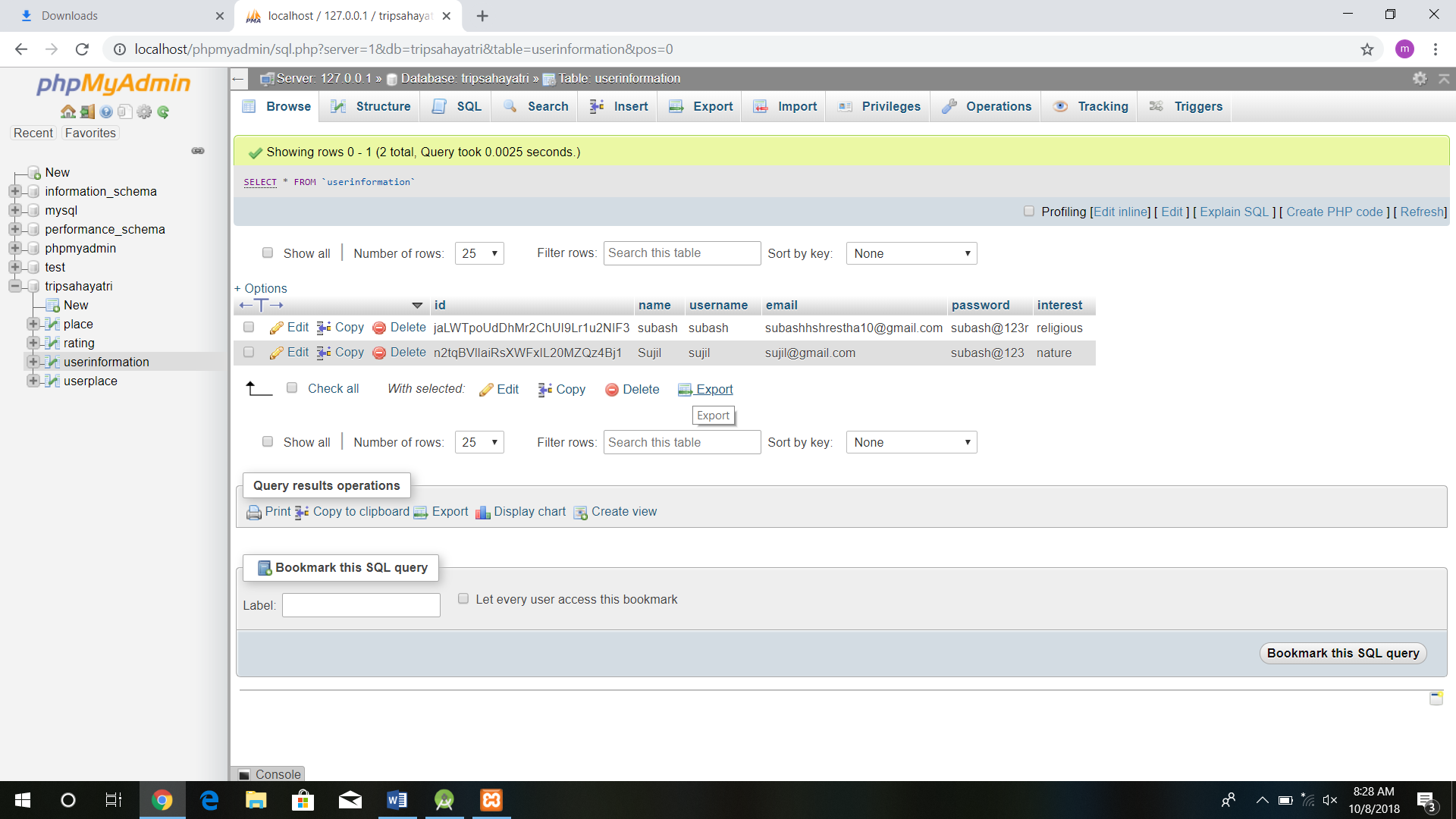
## Survey form used for data collection

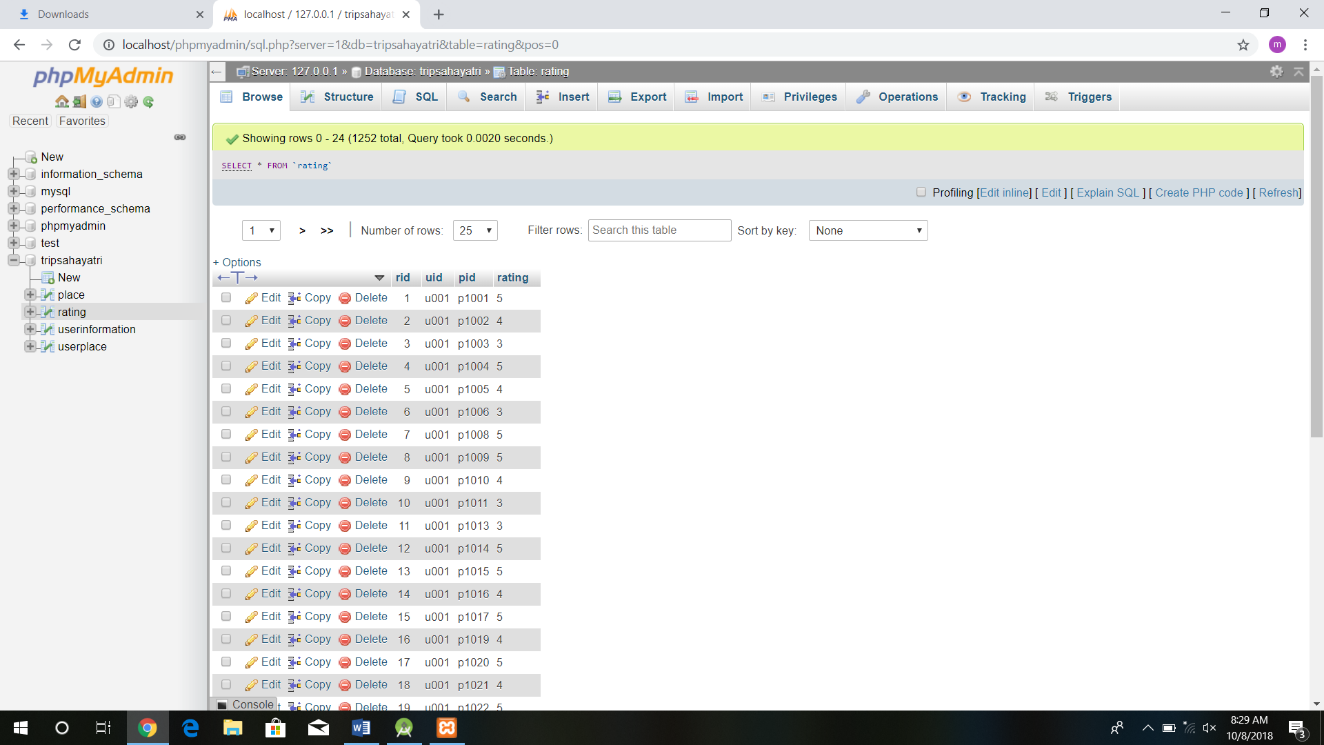


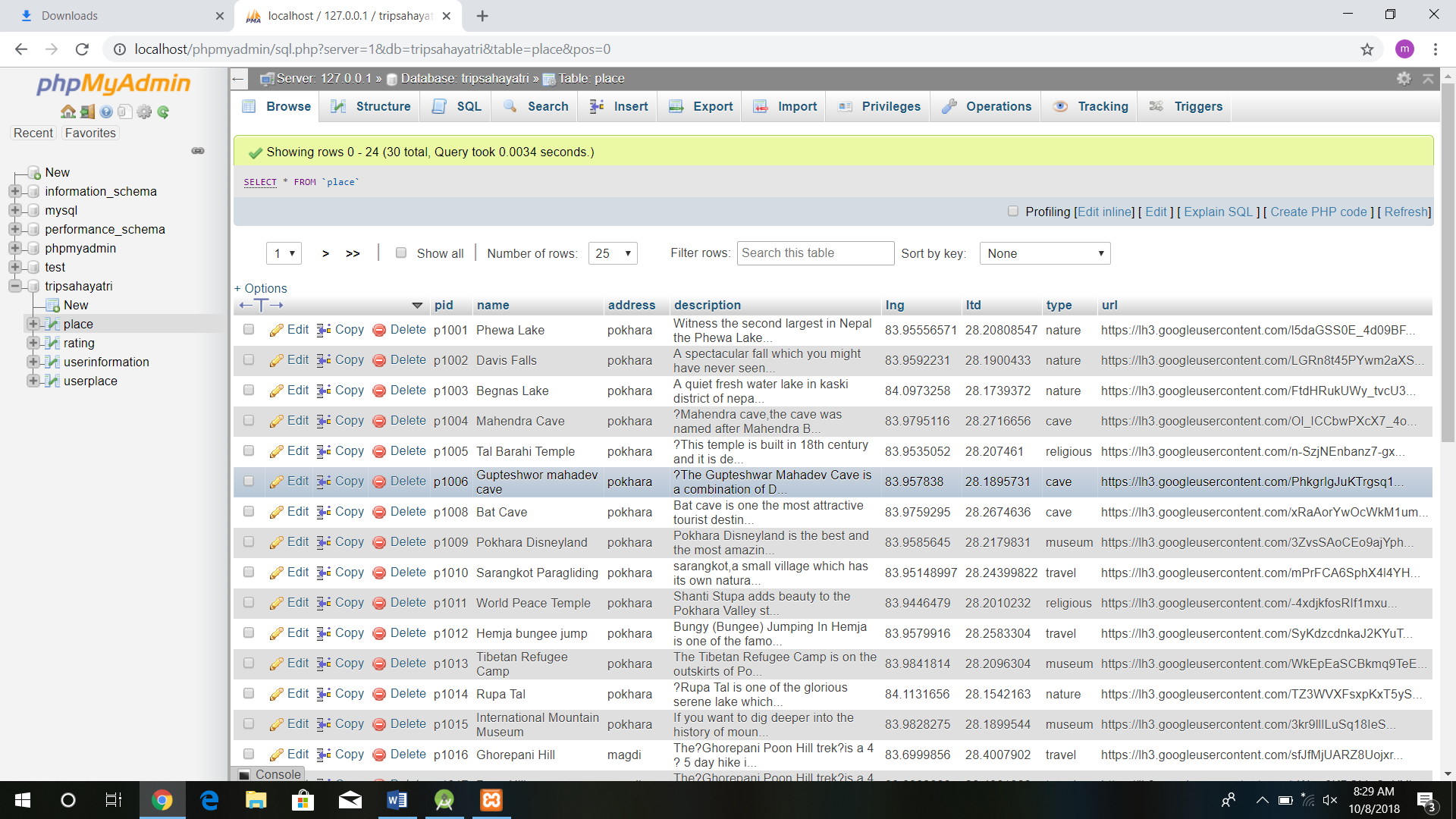
## Database

### 

### 







## 