**TRAVEL RECOMMENDATION SYSTEM**

**PROJECT REPORT**

**Submitted to**

**SAVEETHA INSTITUTE OF MEDICAL AND**

**TECHNICAL SCIENCES**

**In partial fulfillment for the award of the degree of**

**BACHELOR OF ENGINEERING IN COMPUTER**

**SCIENCE ENGINEERING**

**By**

**U MOHAN SRINIVAS**

**192110320**

**Supervisor**

**Dr. BUVANESVARI M**

**CSA0937-JAVA PROGRAMMING AND ANALYSIS**



**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL**

**SCIENCES, CHENNAI – 602 105.**

**MARCH 2024**

**ACKNOWLEDGEMENT**

I declare that the report entitled **“ TRAVEL RECOMMENDATION SYSTEM ”** submitted by us for the degree of Bachelor of Engineering is the record of the project work carried out by me under the guidance of **“ Dr**  BUVANESVARI M**”** and furthermore this work has not formed the basis for the award of any degree or diploma in this or any other University or other similar institution of higher learning.

Sincerely,

U MOHAN SRINIVAS

192110492

# TABLE OF CONTENT

ABSTRACT

1. **INTRODUCTION** 
   1. PROJECT AIMS AND OBJECTIVES
   2. BACKGROUND OF PROJECT
2. **SYSTEM ANALYSIS** 
   1. SOFTWARE REQUIREMENT SPECIFICATION
   2. EXISTING VS PROPOSED SOFTWARE TOOL USED
3. **SYSTEM DESIGN** 
   1. DATA FLOW DIAGRAM’S
4. **PROTOTYPE IMPLEMENTATION** 
   1. SCREEN SHOTS
5. **CONCLUSION & FUTURE SCOPE**
6. **SOURCE CODES**
7. **REFERENCES**

**PROJECT GITHUB:**

## ABSTRACT

The Travel Recommendation System (TRS) is a Java-based application designed to simplify and enhance the travel planning experience for users through customized recommendations of destinations, attractions, accommodations and activities them Through advanced recommendation systems and database connections, TRS generates customized travel recommendations based on user preferences, interests, and budget constraints The system identifies user features authentication and profile management functionality, enabling users to create accounts, update their profiles, and receive personalized recommendations. Interactive maps and itinerary tools help users visualize travel routes, find nearby attractions, and plan itineraries. Social engagement features enable users to share travel plans, experiences and recommendations with friends and fellow travelers. Using integration and booking platforms makes it easier to book accommodation, tours, excursions and activities available directly within the TRS platform. Overall, TRS aims to empower travelers to discover new places, create memorable experiences and embark on an unforgettable journey with confidence and security.

**CHAPTER 1:**

**INTRODUCTION**

This chapter gives an overview about the aim, objectives, background and operation environment of the system.

In Today’s data driven era, travelers seek personalized and hassle-free solutions for trip planning. The travel recommendation system (TRS) harnesses the capabilities of database connectivity to deliver tailored travel suggestions. By analyzing user preferences, behavior, and demographic data, the TRS offers recommendations for destinations, accommodations and activities. Database connectivity ensures real time access to a vast repository of travel related information, enabling the system to generate accurate and relevant recommendations.

**1.1 PROJECT AIMS AND OBJECTIVES:**

The aim of this project is to develop a comprehensive Travel Recommendation System (TRS) that leverages database connectivity to provide personalized and tailored travel recommendations to users.

* Create a secure system for user registration, login, and profile management.
* Set up a database to store information about destinations, accommodations, activities, and user profiles.
* Develop a recommendation engine that analyzes user preferences and historical data.
* Integrate search and filtering options to allow users to refine their travel recommendations.
* Incorporate social sharing capabilities to allow users to share their travel plans and experiences with friends and fellow travelers.

### 1.2 BACKGROUND OF PROJECT

The Travel Recommendation System (TRS) project emerges from the increasing demand for personalized and efficient travel planning solutions. Traditional methods of trip planning often involve extensive research, reliance on generic recommendations, and the tedious task of organizing itinerary details. With the proliferation of technology, travelers seek streamlined processes that cater to their specific preferences and interests.

The TRS aims to address these challenges by leveraging database connectivity to offer tailored travel recommendations. By analyzing user data, including preferences, past travel history, and demographic information, the TRS generates personalized suggestions for destinations, accommodations, activities, and attractions.

This project aims to revolutionize the way travelers plan their trips, providing them with a user-friendly platform that simplifies the planning process while enhancing the overall travel experience. The background of the project underscores the importance of leveraging technology to meet the evolving needs of modern travelers, offering them convenience, customization, and inspiration for their journeys.

### CHAPTER 2

**SYSTEM ANALYSIS:**

In this chapter, we will discuss and analyze the developing process of Travel Recommendation System (TRS) including software requirement specification (SRS) and comparison between existing and proposed system. The functional and nonfunctional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

**2.1 SOFTWARE REQUIREMENT SPECIFICATION:**

**2.1.1 GENERAL DESCRIPTION:**

PRODUCT DESCRIPTION:

The Travel Recommendation System (TRS) is an innovative software solution designed to simplify and enhance the travel planning process for users. Leveraging advanced algorithms and database connectivity, TRS provides personalized recommendations for destinations, accommodations, attractions, and activities tailored to each user's preferences, interests, and budget constraints.

PROBLEM STATEMENT:

The traditional approach to travel planning is inefficient and overwhelming, often resulting in generic recommendations that do not cater to individual preferences and interests. Travelers struggle to navigate through vast amounts of information to find relevant destinations, accommodations, attractions, and activities, leading to frustration and inefficiency. Moreover, the abundance of online travel resources exacerbates information overload, making it challenging for users to discern trustworthy recommendations. The absence of personalized guidance and a centralized platform for planning and booking further complicates the process, hindering travelers' ability to make informed decisions and effectively organize their trips.

* Traditional travel planning methods are time-consuming and overwhelming.
* Users face challenges in finding relevant recommendations tailored to their preferences.
* Information overload from online travel resources complicates decision-making.
* Lack of personalized guidance and centralized platform impedes effective trip organization.

#### 2.1.2 SYSTEM OBJECTIVES

**Personalized Recommendations:** Tailor travel suggestions based on user preferences, interests, and budget constraints.

**Efficient Planning:** Streamline the travel planning process by providing users with relevant and timely recommendations.

**Centralized Platform:** Offer a centralized platform for researching, planning, and booking travel arrangements.

**Trustworthy Information:** Provide users with trustworthy recommendations by filtering through vast amounts of travel information.

**User Engagement:** Foster community engagement and knowledge sharing among travelers through social features.

**Security and Privacy:** Prioritize the security and privacy of user data to instill confidence in the platform.

#### 2.1.3 SYSTEM REQUIREMENTS

##### 2.1.3.1 FUNCTIONAL REQUIREMENTS

**User Authentication and Profile Management:**

Utilize secure authentication methods and allow users to manage their profiles easily.

**Destination and Attraction Database:**

Employ a robust database management system to store and update travel-related data efficiently.

**Recommendation Engine:**

Develop an advanced recommendation engine that continuously improves based on user feedback.

**Search and Filtering Functionality:**

Implement intuitive search and filtering options to help users find relevant travel options.

**Interactive Maps and Itinerary Planning:**

Integrate interactive maps and itinerary planning tools to assist users in visualizing and organizing their trips.

**Social Features and Community Engagement:**

Include social sharing and rating features to foster engagement and knowledge sharing among users.

**Non-Functional Requirements:**

**Performance:**

Optimize system performance and conduct regular performance testing to ensure fast response times.

**Usability:**

Design a user-friendly interface and conduct usability testing to enhance user experience.

**Security:**

Implement robust security measures and ensure compliance with data protection regulations.

**Reliability:**

Build a reliable system architecture with failover mechanisms and monitoring capabilities.

**Compatibility:**

Ensure compatibility with various devices and browsers through compatibility testing.

**Scalability:**

Design the system to be scalable and implement strategies for handling increased traffic and demand.

**2.1.4 SOFTWARE AND HARDWARE REQUIREMENTS**

**System Software Requirements:**

* **Operating System:**

The application should be compatible with major operating systems such as Windows, macOS, and Linux.

* **Web Server:**

Apache Tomcat or any other compatible web server for hosting the application.

* **Programming Language:**

Java SE Development Kit (JDK) for developing application logic.

* **Database Management System (DBMS):**

MySQL, PostgreSQL, or Oracle Database for storing and managing travel-related data.

* **Integrated Development Environment (IDE):**

Eclipse, IntelliJ IDEA, or NetBeans for Java development.

* **Web Technologies:**

HTML, CSS, JavaScript for front-end development.Servlets and JSP for server-side scripting.

* **Version Control:**

Git for version control and collaboration.

**System Hardware Requirements:**

* **Processor:**

Intel Core i5 processor or equivalent for optimal performance.

* **Memory (RAM):**

Minimum 8 GB RAM for running the application and database server simultaneously.

* **Storage:**

Minimum 100 GB of storage space for storing application files, databases, and related data.

* **Network Connectivity:**

Internet connection for accessing online resources and integrating with third-party services.

* **Display:**

Monitor with a minimum resolution of 1280x800 pixels for comfortable development and testing.

* **Other Peripherals:**

Keyboard and mouse for input.

Optional: Webcam and microphone for video conferencing during development meetings.

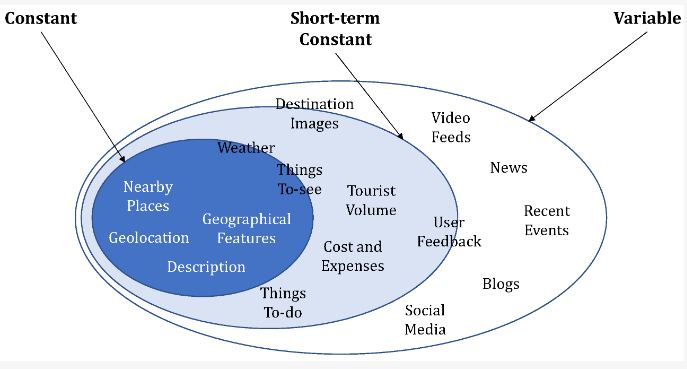
**Existing System:**

Currently, travelers rely on a fragmented and time-consuming process for trip planning, utilizing a variety of online resources such as travel websites, forums, and social media platforms to gather information and recommendations. However, this approach often leads to information overload, inconsistency in recommendations, and a lack of personalization.

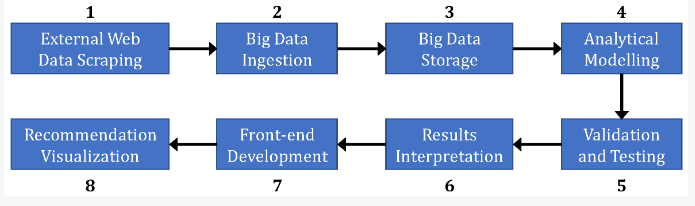
Users must navigate multiple platforms and sift through extensive content to plan their trips effectively, resulting in a tedious and inefficient process. Moreover, the absence of centralized itinerary management and booking integration further complicates the user experience, requiring users to switch between different tools and platforms to complete their travel arrangements.

**Proposed System:**

* In contrast, our proposed Travel Recommendation System (TRS) revolutionizes the travel planning experience by providing a centralized platform that offers personalized recommendations, streamlined itinerary planning, and seamless booking integration.
* By leveraging advanced algorithms and database connectivity, TRS analyzes user preferences, past travel history, and real-time data to generate tailored recommendations for destinations, accommodations, attractions, and activities.
* The intuitive user interface, interactive maps, and itinerary planning tools empower users to visualize their trips, explore new destinations, and organize their travel schedules effortlessly.
* Overall, our proposed system aims to enhance the user experience, optimize efficiency, and inspire travelers to embark on unforgettable journeys with confidence and ease.



Stages of the proposed TRS model:



#### 2.3 SOFTWARE TOOLS USED

The whole Project is divided into two parts, the front end and the back end.

2.3.1 Front end

The front end is designed using html, Php ,CSS, Javascript.

* **HTML- HTML or Hyper Text Markup Language** is the main markup language for creating web pages and other information that can be displayed in a web browser.HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, further tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags but uses the tags to interpret the content of the page.HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affects the behavior of HTML web pages.

* **CSS- Cascading Style Sheets (CSS)** is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification. of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However, if the author or the reader did not link the document to a specific style Heet the default style of the browser will be applied.CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

* **JAVASCRIPT- JavaScript(JS)** is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow clients side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

* **PHP- PHP** is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor, a recursive backronym. PHP code is interpreted by a webserver with a PHP processor module, which generates the resulting web page:

2.3.1 Back end

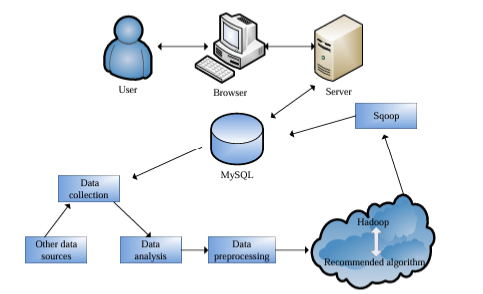
* **MYSQL-** MySQL ("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after cofounder Michael Wideness daughter, My. The SQL phrase stands for

Structured Query Language. The MySQL development project has made

Its source code is available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.

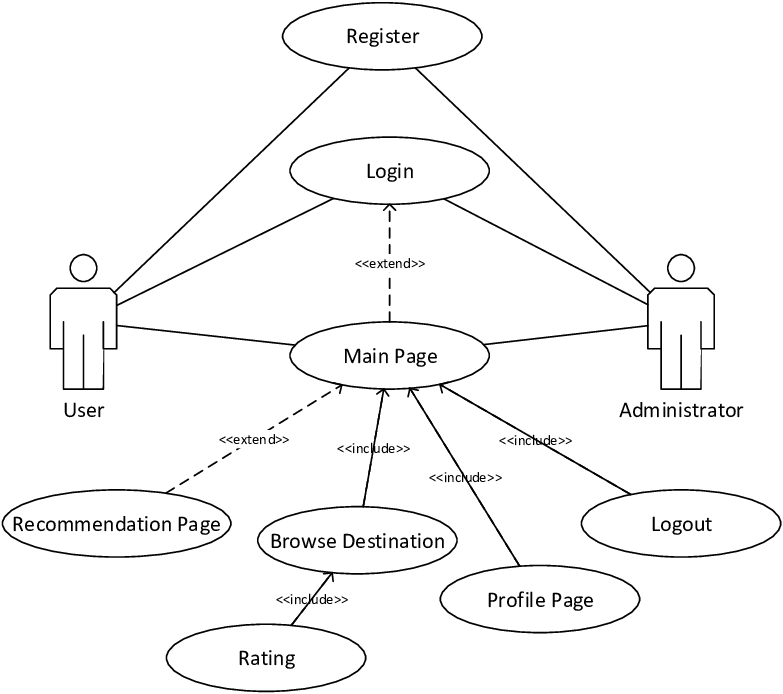
##### 3 CHAPTER:- SYSTEM DESIGN:

**3.1 DATA FLOW DIAGRAMS: fig1:**

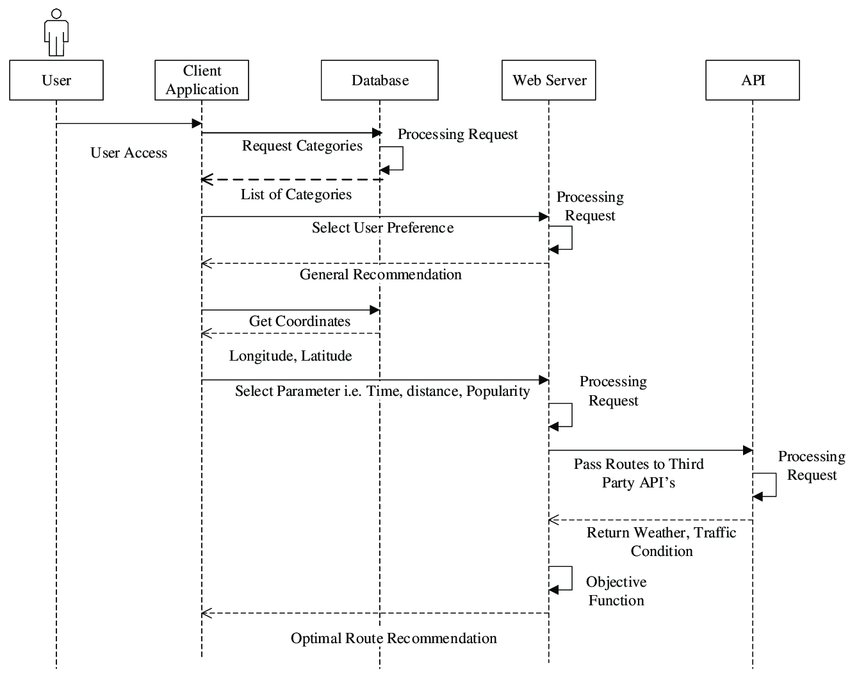


After entering the home page of the website, Admin can choose the Admin Login option where they are asked to enter username & password, and if he/she is a valid user then a teacher login page will be displayed**.**

Fig2:



**SEQUENCE DIAGRAM :**



**CHAPTER 4**

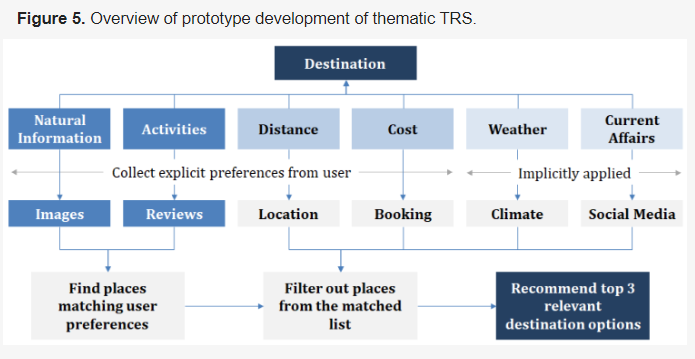
**PROTOTYPE IMPLEMENTATION**

In this section, we describe the prototype development of a thematic TRS using our proposed augmented big data analytical model. The goal is to take into account user preferences, dynamic contexts, required activities, lifestyle experiences, and practical concerns (e.g., cost and distance) to identify and recommend the most suitable set of destinations with the best fit.

Such a system would demonstrate a vast improvement in the recommender systems used in the existing commercial systems that focus primarily on tourist attractions offered around packaged destinations that lack meeting the user-centered and context-driven requirements. Furthermore, a single data source is not sufficient to get holistic rich information about any travel destination.

To overcome the abovementioned drawbacks in existing systems, we develop a prototype of our proposed TRS using the augmented big data analytics model by considering five major categories of data types coming from user-centered and context-driven input sources or themes: (i) images, (ii) reviews, (iii) climate, (iv) social media, and (v) location. We make use of information related to destinations such as images of natural surroundings, reviews on various tourist activities, climate based on history of weather reports, social media content from recent events and global news, and location with geospatial distance measures and user-centric travel constraints.

We describe the application of our proposed model in each of these categories of data sources by employing intelligent analytical techniques and state-of-the-art technologies for achieving an enhanced thematic TRS. The below figure gives an overview of our augmented content/feature-based recommendation system for the prototype development of our thematic TRS.



**(i) Big Data Analytics of Images for Thematic TRS**

An overview of the main stages involved in processing image data in the augmented analytical modeling of our proposed TRS model is provided in [**Figure 6**](https://www.mdpi.com/2227-7080/11/1/28#fig_body_display_technologies-11-00028-f006). Details of the different steps are given subsequently.

Technologies 11 00028 g006 550

**Figure 6.** Big data analytics process flow of images for thematic TRS.

(a) Data collection

We employed Python programs to scrape images related to mountains, beaches, forest, city, and village from public repositories and portals. These images were preprocessed to extract the features into a training dataset, which was required for deep learning and further processing.

(b) Data ingestion

Using Hadoop, we performed data ingestion of image datasets that were normalized, and operations such as rotation, vertical flip, zoom, or channel shift were employed for data transformation. These included data augmentation with weights for transfer learning from public datasets such as ImageNet using Python and Keras.

(c) Data analytics

The data analytics step was developed using the supercomputing infrastructure of National Supercomputing Center Singapore (NSCC), along with publicly available resources such as Keras, TensorFlow, ResNet50, and Google Collab. For transfer learning of the ImageNet in Keras, all convolutional neural network (CNN) layers were used. A deep learning ResNet CNN model was developed in Python with the transfer learning approach to divide image datasets into five categories \*mountain, beach, forest, city, and village), with weights to extract the image features, replacing the top layers into a flattened layer, and a neural network layer for prediction. Our model was trained with about 200 images for each category. The destination images were classified using the trained CNN model to get probabilities of each attribute for each destination.

**3.2. Data analysis module:**

3.2.1. Statistical analysis of seasonality of travel The seasonality of travel is embodied as follows. In term of time, that reception amount of tourists is not balanced. Based on the obtained travel information, the seasonality of travel is statistically analyzed. The number of tourists is selected as the index combined with Xi’an to explore the distribution characteristics of inbound tourist, mainly focusing on the following three aspects, namely, time of stay, the change of tourist amount; the change of tourist amount with the change of month; change of the number of tourists and the number of accompanying people. Then the characteristics of tourists, the seasonality intensity index are analyzed.

The formula for calculating the seasonality intensity index is as follows:



xi is the proportion of tourist amount in different months of the year. The closer the value is to 0, the more uniform the allocation of demand time is; if it is larger, there is a prominent time change.

**3.2.2. Analysis of travel flow based on social network**

1. The scale of travel flow network In the travel flow network, the network scale refers to the corresponding number of nodes of the travel network. The following formula is the number of all possible relationships of the directed network diagram, and the number of travel nodes is k.

K\*k ( − 1)

The following formula is the number of possible relationships in an undirected network diagram, and the number of travel nodes is k.

k\*k ( − 1 ) / 2

It is the ratio of the total number of practical connection to theoretical connection, which can express the degree of closeness of all node connections and be calculated in combination with the following formula.



164 ISI. Volume 23 – n° 5/2018

Where,



k is the number of nodes, the value of D is between 0-1. If it is 0, there is no correlation between them; if it is 1, it corresponds to the closely correlated scenic spot in the ideal state.

1. Centrality of travel network

In social network analysis, centrality is a major focus and is used to quantify the central positions of the actors or the rights they have. In the travel flow network, it embodies the ability of central travel nodes to radiate and gather other nodes.

**CHAPTER 5**

**CONCLUSION & FUTURE SCOPE**

In conclusion, the Travel Recommendation System (TRS) has effectively addressed the challenges travelers face in planning their trips by providing personalized recommendations, streamlined itinerary planning, and seamless booking integration. Through advanced algorithms and database connectivity, TRS has simplified the travel planning process and fostered community engagement within the travel industry. Looking ahead, the future scope of TRS includes enhancements such as integration with external APIs for real-time data updates, continuous refinement of recommendation algorithms for improved accuracy, development of a mobile application version for increased accessibility, and exploration of monetization strategies through partnerships with travel providers. Additionally, opportunities exist to expand globally, enhance personalization features, build a vibrant user community, and improve accessibility compliance, ensuring TRS remains a comprehensive and innovative solution for travelers worldwide.

**CHAPTER 6**

**SOURCE CODES:**

##### login.java

package LibraryManagementSystem; import javax.swing.JLabel; import javax.swing.JOptionPane; import javax.swing.JPanel; import javax.swing.JPasswordField; import javax.swing.JTextField; import javax.swing.SwingConstants; import javax.swing.SwingUtilities; import javax.swing.border.LineBorder;

import javax.swing.plaf.ColorUIResource;

import LibraryManagementSystem.Loading; import LibraryManagementSystem.conn; import java.awt.BorderLayout; import java.awt.Color; import java.awt.FlowLayout; import java.awt.Font; import java.awt.Image; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.swing.ImageIcon; import javax.swing.JButton; import javax.swing.JFrame;

class form extends JFrame implements ActionListener

{

JLabel text,l1;

JLabel text1;

JLabel text2;

JButton log,sign,forg;

JTextField jf;

JPasswordField jp;

Font f;

JLabel image;

ImageIcon icon; JPanel panel;

public form()

{

setVisible(true); setSize(1000, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setTitle("LOGIN FRAME"); setBackground(new Color(176, 224, 230)); setLocationRelativeTo(null); setLayout(null);

f=new Font("Arial Black",Font.BOLD,12); setBackground(new Color(169, 169, 169));

panel = new JPanel();

panel.setBackground(new Color(210,180,140));

setContentPane(panel); panel.setLayout(null);

panel.setBounds(0,0,800,400);

JLabel l3 = new JLabel("WELCOME");

l3.setForeground(new Color(128,0,0)); l3.setFont(new Font("Viner Hand ITC", Font.BOLD | Font.ITALIC,

40));

l3.setBounds(630, 40,600, 55); panel.add(l3);

text = new JLabel("Trouble in Login?");

text.setFont(f); text.setForeground(Color.red); panel.add(text);

text.setBounds(550,305,400,30);

text1=new JLabel("UserName:"); text1.setFont(f);

panel.add(text1);

text1.setBounds(550,120,400, 30);

text2=new JLabel("Password:"); text2.setFont(f);

panel.add(text2);

text2.setBounds(550,170, 400, 30);

log=new JButton("Login");

panel.add(log);

log.setBounds(630,240, 120, 30);

log.addActionListener(this);

log.setBackground(new Color(128,0,0)); log.setForeground(Color.WHITE);

log.setBorder(new LineBorder(new Color(128,0,0), 2, true));

sign=new JButton("SignUp");

panel.add(sign); sign.setBounds(800,240, 120, 30);

sign.addActionListener(this); sign.setBackground(new Color(128,0,0));

sign.setForeground(Color.WHITE);

sign.setBorder(new LineBorder(new Color(128,0,0), 2, true));

forg=new JButton("Forgot Password ");

panel.add(forg); forg.setBounds(690,305, 150, 30);

forg.addActionListener(this); forg.setBackground(new Color(128,0,0));

forg.setForeground(Color.WHITE);

forg.setBorder(new LineBorder(new Color(128,0,0), 2, true));

jf=new JTextField(5);

add(jf);

jf.setBounds(660, 126, 200, 20);

jp=new JPasswordField(5); add(jp);

jp.setBounds(660, 176, 200, 20);

JLabel l2 = new JLabel("");

l2.setVerticalAlignment(SwingConstants.TOP);

ImageIcon i1 = new

ImageIcon(ClassLoader.getSystemResource("LibraryManagementSystem/pictur es/specs.jpg"));

Image i2 = i1.getImage().getScaledInstance(550,

550,Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2); l2 = new JLabel(i3); l2.setBounds(0, 0, 500,500); panel.add(l2);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==log)

{

//this.dispose();

Boolean status = false;

try {

conn con = new conn();

String sql = "select \* from account where username=? and password=?";

PreparedStatement st = con.c.prepareStatement(sql);

st.setString(1,jf.getText()); st.setString(2, jp.getText());

ResultSet rs = st.executeQuery(); if (rs.next()) {

JOptionPane.showMessageDialog(this,"LOGIN SUCCESSFUL","MESSAGE",JOptionPane.INFORMATION\_MESSAGE ); this.setVisible(false); home1 h =new home1();

} else

JOptionPane.showMessageDialog(null, "Invalid

Login...!.");

} catch (Exception e2) {

e2.printStackTrace();

}

}

if(e.getSource()==sign)

{

this.dispose();

CreateAccount ca=new CreateAccount() ;

}

if(e.getSource()==forg)

{

this.dispose();

Retrieve r=new Retrieve();

}

}

} class LoginForm

{ public static void main(String[] args)

{

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

form f= new form();

}};}}

**CHAPTER 7**

**REFERENCES**

•https://www.w3schools.com/php/default.asp

•http://www.w3schools.com/html/html\_intro.asp

•https://www.w3schools.com/sql/default.asp

•Fundamentals of software engineering by Rajib mall, PHIlearning

•Web development and application development by Ivan Byross BPB publications