Recipe Recommendation Application

Minor Project Report

Submitted for the partial fulfillment of the degree of

Bachelor of Technology

In

Internet of Things (IOT)

Submitted By

Yash Gupta
0901IO221077

UNDER THE SUPERVISION AND GUIDANCE OF

Dr. Priyanka Garg
Assistant Professor

Centre for Internet of Things

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA
माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत
Deemed to be university
NAAC ACCREDITED WITH A++ GRADEA
Jul-Dec 2024

DECLARATION BY THE CANDIDATE

I hereby declare that the work entitled **"Recipe Recommendation Application"** is my work, conducted under the supervision of **DR. Priyanka Garg,** during the session Jan-May 2024. The report submitted by me is a record of bonafide work carried out by me.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

19/11/24

Yash Gupta
0901IO221077
B.Tech. V Sem

Date: 19 Nov 2024
Place: Gwalior

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Guided By:

Priyanka 19/11/24

Dr. Priyanka Garg
Assistant Professor
Center for Internet of Things
MITS, Gwalior

Departmental Project Coordinator

Dr. Nookala Venu
Assistant Professor
Centre for Internet of Things
MITS, Gwalior

Approved by HoD

19/11/24

Dr. Praveen Bansal
Assistant Professor
Centre for Internet of Things
MITS, Gwalior

PLAGIARISM CHECK CERTIFICATE

This is to certify that I/we, a student of B.Tech. in **Internet of Things (IOT)** have checked my complete report entitled **"Recipe Recommendation Application"** for similarity/plagiarism using the "Turnitin" software available in the institute.

This is to certify that the similarity in my report is found to be 16% which is within the specified limit (20%).

The full plagiarism report along with the summary is enclosed.

14/11/24

Yash Gupta

0901IO221077

Checked & Approved By:

Soumyajit Ghosh 20/11/24

Dr. Soumyajit Ghosh
Assistant Professor
Centre for Internet of Things
MITS, Gwalior

ABSTRACT

Recipe Recommendation Application is a mobile application built in Flutter, integrated with Firebase and the Spoonacular API. This app is to help users discover recipes and suggest what meal can you have with the ingredients. The app allows users to enter ingredients they have available and returns live recipe suggestions from the Spoonacular API. It also allows users to save their favorite recipes for easy access later.

The app has a very clean user interface and is simple to use, so it can be used by every age group of people. Firebase is used to securely authenticate users and store their data using it on a secure server with privacy-aware reliability. Working with APIs and cloud services, the app connects technology to a task as everyday as meal preparation.

This project shows the power of mobile apps to make everyday life easier and avoid real-world inconveniences. It could further enhance by offline mode, meal planning, and dietary preference support in the future.

ACKNOWLEDGEMENT

The full semester I Project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology & Science** to allow me to continue my disciplinary/interdisciplinary Internship/ Project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Internet of Things,** for allowing me to explore this project. I humbly thank **Dr. Praveen Bansal**, Assistant Professor and Coordinator, Centre for Internet of Things, for his continued support during the course of this engagement, which eased the process and formalities involved. I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Dr. Priyanka Garg**, Assistant Professor, and Centre for Internet of Things, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

19/11/24

Yash Gupta

0901IO221077

Centre for Internet of Things

CONTENT

Table of Contents

ACRONYMS

- **API** - Application Programming Interface
- **UI** - User Interface
- **UX** - User Experience
- **Firebase** - Firebase is a platform for developing mobile and web applications
- **SDK** - Software Development Kit

NOMENCLATURE

□ **Flutter**: A UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase.

□ **Firebase**: A platform developed by Google for building mobile and web applications, providing features like authentication, real-time databases, and cloud storage.

□ **API**: Application Programming Interface – A set of tools and protocols used to build software and interact with external services or data sources.

□ **Spoonacular API**: A web service that provides access to food-related data, such as recipes, ingredients, and nutritional information.

□ **Firebase Authentication**: A service that helps you authenticate users using only client-side code.

□ **Favorites**: A section in the app where users can save and access recipes they like for later use.

□ **Cloud Firestore**: A flexible, scalable database for storing and syncing data in real-time, used in Firebase.

□ **UI**: User Interface – The space where interactions between users and the application occur, designed to be user-friendly.

□ **UX**: User Experience – The overall experience a user has when interacting with the app, focusing on ease of use and satisfaction.

Chapter 1: Introduction

The **Recipe Recommendation App** is a mobile application designed to help users find meal ideas based on the ingredients they have at home. Many people struggle with deciding what to cook, especially when they only have a few ingredients available. This app solves that problem by providing recipe suggestions using the **Spoonacular API**, which offers a wide range of recipes based on user inputs.

Built with **Flutter**, a cross-platform framework, this app works on both Android and iOS devices, offering a smooth user experience on both platforms. **Firebase** is used for backend services like user authentication, storing data, and real-time updates. The app allows users to log in, enter the ingredients they have, get recipe suggestions, and save their favorite recipes for later.

This project demonstrates how mobile development, cloud-based services, and external APIs can work together to create a useful and practical solution for everyday tasks like meal planning. Key features of the app include the ability to suggest recipes based on ingredients, authenticate users, and save favorite recipes. The project highlights how modern technologies can be used to provide real-time, personalized recommendations for users.

The following chapters will explore the design, development, and features of the app, as well as potential improvements that could be made in the future.

CHAPTER 2: LITERATURE SURVEY

This a mobile application called the Recipe Recommendation App that helps you to find meal ideas according to ingredients available at your home. Deciding what to make for dinner can be especially difficult at times, but with only a few ingredients on hand, sometimes it becomes impossible. Now, this app tackles that problem and suggests recipes to you using the Spoonacular API, which has millions of recipes based on user inputs [1][2].

This app is developed in Flutter, which is a cross-platform framework, so both Android and iOS devices will get the same user experience without any discrepancies [3]. Firebase is used to provide backend services such as user authentication, data storage, and real-time updates [4]. Users can log in, input what they have, get recipe ideas, or save the recipes for later.

The project illustrates the integration of mobile development, cloud-based services as a dynamic and interactive solution to perform everyday tasks such as meal planning. Some of the major features that my app offers are ingredient-based recipe suggestions along with authentication for the user to save favorite recipes [5]. This project emphasizes the usage of modern technology to provide real-time, customized suggestions for users [6].

The literature survey focuses on the approaches used in similar applications and how they have been implemented in the context of food recommendation, ingredient-based searches, and the use of external data sources for personalized recommendations [2][7].

1. Recipe Recommendation Systems

Several studies and applications have been developed to provide recipe recommendations based on various parameters, such as ingredients, dietary preferences, and nutritional information. A common approach in recipe recommendation systems is the use of collaborative filtering, where recommendations are based on the preferences and ratings of other users. However, ingredient-based recommendation systems, which allow users to input available ingredients to find recipes, are becoming in…

The survey of existing literature concentrates on methods that relevant work has (or within reason) been adopted rather than specifically being implemented in these three areas: food recommendation, ingredient-based searches, and the utilization of external sources to inform personalized recommendations.

1. Recipe Recommendation Systems

Several studies and applications have been proposed to generate recipes according to ingredient(s), dietary requirements, and nutritional values. One of the most widely used techniques in recipe recommendation systems is collaborative filtering, which predicts ratings for a user based on the preferences or ratings of other users. However, ingredient-based recommendation systems, where you can enter your available ingredients and find recipes, are also gaining popularity since they represent more natural use cases in our daily meals.

In 2019, Liu et al. implemented machine learning algorithms based on ingredients, preferences, and user past interactions to develop a recipe recommendation system. Trained on a large recipe dataset, the system successfully matched users to meals they were likely to enjoy based on previous selections.

Zhao et al. proposed another approach in 2021, emphasizing recipe recommendations based on dietary constraints like vegetarian, gluten-free, or low-carb diets. This model not only took

1. Recipe Recommendation Systems

Several studies and applications have been developed to provide recipe recommendations based on various parameters, such as ingredients, dietary preferences, and nutritional information. A common approach in recipe recommendation systems is the use of collaborative filtering, where recommendations are based on the preferences and ratings of other users. However, ingredient-based recommendation systems, which allow users to input available ingredients to find recipes, are becoming in…

The survey of existing literature concentrates on methods that relevant work has (or within reason) been adopted rather than specifically being implemented in these three areas: food recommendation, ingredient-based searches, and the utilization of external sources to inform personalized recommendations.

1. Recipe Recommendation Systems

Several studies and applications have been proposed to generate recipes according to ingredient(s), dietary requirements, and nutritional values. One of the most widely used techniques in recipe recommendation systems is collaborative filtering, which predicts ratings for a user based on the preferences or ratings of other users. However, ingredient-based recommendation systems, where you can enter your available ingredients and find recipes, are also gaining popularity since they represent more natural use cases in our daily meals.

In 2019, Liu et al. implemented machine learning algorithms based on ingredients, preferences, and user past interactions to develop a recipe recommendation system. Trained on a large recipe dataset, the system successfully matched users to meals they were likely to enjoy based on previous selections.

Zhao et al. proposed another approach in 2021, emphasizing recipe recommendations based on dietary constraints like vegetarian, gluten-free, or low-carb diets. This model not only took into account available ingredients but also filtered such suggestions with dietary preferences and health goals, offering more personalized recipe options.

2. Flutter-Based Mobile Application Development

With the introduction of a cross-platform mobile app development framework like Flutter, developing mobile apps for both Android and iOS has become way easier than before. A study by Smith et al. in 2020 compared Flutter with other frameworks like React Native and Native app development to showcase the benefits of using Flutter over them. Due to this popularity, many e-commerce applications, social media apps, and health and food-related apps have adopted this framework.

When it comes to food and recipe apps, Flutter's fast development cycles with smooth UX implementation for all platforms make it an ideal choice. According to a review by Johnson (2021), Flutter is widget-based, which makes UI development much quicker, and many apps designed with it are visually appealing and easy to use.

3. API Integration in Mobile Applications

Another key area of modern app development is using APIs (Application Programming Interfaces) for mobile application enhancement. With third-party API integrations, developers can add additional functionality to their apps without having to implement complex features from scratch. The Spoonacular API used in this project is one of the more popular APIs for working with recipe data, providing access to a large amount of recipes, ingredients, nutritional data, and more.

A study by Miller et al. in 2019 investigated public APIs used in mobile apps and showed how APIs like Spoonacular provide real-time data for different categories to add variety to app functionality. They discussed how APIs provide developers with ready-made solutions, helping save time and resources while implementing apps that require large amounts of data, such as recipe recommendation apps.

4. Mobile App Backend: Firebase

Firebase is a Google platform widely used for mobile app development to manage backend services like authentication, real-time databases, cloud storage, and hosting.

According to Brown et al. (2020), Firebase offers real-time database and user authentication services, making them easy to use for applications that require real time data syncing and secure management of users. For example, Firebase Authentication takes care of allowing users to easily sign in, while Firestore database can be used to store user data and save favorite recipes.

5. Trends in Recipe Recommendation Apps

In a recent review by Williams (2022), new areas of research and interest in recipe recommendation apps are discussed, with personalization becoming one of the highest demands in daily life. Today, recipe apps don't just look at what is lying around your kitchen, they know about your dietary restrictions, health goals, and even previous choices.

Also gaining significant attention is food sustainability, as many apps offer ideas on recipes that can avoid wasting perishable ingredients. For example, applications like Yummly and SuperCook incorporate this concept, enabling users to search based on the food they already have at home, reducing waste.

CHAPTER 3: SYSTEM REQUIREMENTS

The system requirements for the **Recipe Recommendation App** can be categorized into two parts: software requirements and hardware requirements. These specifications ensure the app functions smoothly during development and end-user operation.

1. Software Requirements

To develop, test, and deploy the app, the following software tools and technologies are required:

- **Operating System**: Windows 10 or later, macOS 10.14 or later, or a Linux distribution (Ubuntu 20.04 or similar).
- **IDE**: Android Studio or Visual Studio Code with the Flutter and Dart plugins for application development.
- **Programming Language**: Dart (for Flutter development).
- **Framework**: Flutter SDK (version 3.0 or later).
- **Backend Services**: Firebase for authentication, database management, and real-time storage.
- **API**: Spoonacular API for fetching recipe data based on ingredients.
- **Browser**: Chrome or equivalent, for testing web versions and debugging.
- **Version Control**: Git, for managing code versions and collaborating effectively.
- **Additional Tools**: Postman for testing API endpoints and Google Maps SDK (if integrated).

2. Hardware Requirements

The following hardware specifications are recommended for development and testing:

- **Development Machine**:
  - **Processor**: Intel Core i5 or higher / AMD Ryzen 5 or higher.
  - **RAM**: 8 GB (minimum), 16 GB (recommended for smoother performance).
  - **Storage**: 500 GB HDD or 256 GB SSD (recommended for faster operations).
  - **Graphics**: Integrated or dedicated graphics card for emulation and testing.
  - **Display**: 1080p resolution (minimum) for better IDE visibility.
- **Test Devices**:
  - **Smartphones**: Android (version 9.0 or later) and iOS (version 13.0 or later) devices for compatibility testing.
  - **Emulators**: Android Emulator and iOS Simulator for testing different screen sizes and resolutions.

3. Network Requirements

Since the app relies on external services like Firebase and Spoonacular API, a stable internet connection is essential:

- **Speed**: Minimum of 10 Mbps for downloading required SDKs, APIs, and testing.
- **Latency**: Low latency is preferred to ensure smooth API communication.

These system requirements ensure seamless app development, testing, and operation. They also provide a reliable foundation for integrating features like real-time database syncing, secure login, and API-based recipe recommendations. This setup supports efficient performance during both the development and deployment phases.

CHAPTER 4: SYSTEM DESIGN

The system design of the Recipe Recommendation App outlines the architectural structure, data flow, and key components required to ensure seamless operation. This section provides a high-level overview of how the system's components interact to deliver the app's core functionalities, such as user authentication, recipe recommendations, and data storage.

1. Architectural Overview

The app follows a **client-server architecture**, where the client-side (Flutter application) interacts with the server-side components (Firebase and Spoonacular API) to fetch, process, and store data. The architecture ensures scalability, real-time synchronization, and secure data handling.

- **Frontend**: Flutter framework is used to build the app's user interface, ensuring compatibility across Android and iOS platforms.
- **Backend**: Firebase serves as the backend to handle authentication, database management, and real-time operations.
- **External API**: The Spoonacular API is integrated to fetch recipe data based on user inputs.

2. Component Design

2.1. User Interface (UI)

- The app interface is designed using **Flutter widgets**, enabling a responsive and interactive user experience.
- Key UI components include:
    - **Login Screen**: Allows users to register or sign in by using Firebase Authentication.

  - **Ingredient Input Screen**: Users can enter ingredients manually or select them from a list.
  - **Recipe Results Screen**: Displays a list of suggested recipes fetched from the API.
  - **Favorites Screen**: Enables users to save and view their favorite recipes.

2.2. Backend System

- **Firebase Authentication**:
  - Manages user login and registration securely using email and password or social login methods.
- **Firestore Database**:
  - Stores user data, including saved recipes and user preferences, in a structured and scalable format.
- **Firebase Cloud Functions**:
  - Hand!es server-side logic, such as validating API calls and managing complex database queries.

2.3. API Integration

- The Spoonacular API is integrated to fetch recipe data dynamically based on user-provided ingredients.
- API functionalities include:
  - Fetching recipe details such as ingredients, preparation steps, and nutritional information.
  - Filtering recipes based on dietary preferences or restrictions.

3. Data Flow Diagram (DFD)

Level 0: High-Level Overview

1. User inputs ingredients into the app.
2. The app sends a request to the Spoonacular API to fetch recipes.
3. Recipe data is displayed to the user.
4. Users can save their favorite recipes, which are stored in the Firebase Firestore database.

Level 1: Detailed Flow

1. **User Input**:
   - Users provide input via the Ingredient Input Screen.
2. **Data Processing**:
   - The app processes the input and sends a query to the Spoonacular API.
3. **API Response**:
   - The API returns a list of matching recipes, which is displayed to the user.
4. **Database Storage**:
   - When a recipe is marked as a favorite, it is saved to the Firestore database under the user's account.
5. **User Authentication**:
   - Firebase handles login, registration, and session management for secure access.

4. System Interaction

- **User Interaction**:
  - Users interact with the app through the intuitive UI, which includes forms, buttons, and dynamic lists.
- **Server Interaction**:
  - The app communicates with Firebase and the Spoonacular API to retrieve and store data in real-time.
- **Error Handling**:
  - The app provides error messages for issues such as network failure, invalid API responses, or authentication errors.

---

5. Database Design

The Firebase Firestore database is used to store user-related data:

- **Users Collection**:
  - Contains user profiles with fields like userId, email, and passwordHash.
- **Favorites Collection**:
  - Stores favorite recipes with fields like recipeId, title, ingredients, and userId (for mapping).

---

6. Security Measures

- **Authentication**:
  - Firebase Authentication ensures only authorized users access the app.

- **Data Encryption**:
  - All communication between the app, Firebase, and Spoonacular API is encrypted using HTTPS.
- **Access Control**:
  - Firestore rules are configured to restrict access to user-specific data.

This system design ensures the app is modular, secure, and scalable, enabling it to provide a seamless user experience while efficiently managing resources.

CHAPTER 5: IMPLEMENTATION

The Recipe Recommendation App was implemented using Flutter for the frontend, Firebase for authentication and database management, and the Spoonacular API for fetching recipe data. The development process involved integrating these technologies seamlessly to create an intuitive and efficient application.

Frontend Development

The user interface was built with Flutter, which provides a responsive and cross-platform experience. Screens were designed to allow users to input ingredients, view recipe suggestions, manage favorites, and handle user authentication. Essential UI elements like forms, lists, and navigation bars were implemented to ensure smooth navigation across the app.

Backend Integration

Firebase was integrated as the backend to manage user authentication and store user-specific data such as favorite recipes. Firebase Authentication enabled secure login and registration, while Firestore served as a real-time database for storing user preferences.

API Integration

The Spoonacular API was used to fetch recipe suggestions based on ingredients provided by the user. The app sends API requests and processes responses to display detailed recipe information, including ingredients, preparation steps, and nutritional data. Proper error handling was implemented to manage scenarios like network failures or invalid inputs.

Data Flow and Functionality

1. Users input their ingredients, which are processed and sent to the Spoonacular API.

2. The API responds with a list of recipes, which are displayed on the results screen
3. Users can mark recipes as favorites, which are then stored in Firebase Firestore for future access.

Testing and Deployment

The app was rigorously tested to ensure it worked smoothly on various devices and handled edge cases effectively. After debugging and optimization, the final build was prepared and deployed for use

CHAPTER 6: TESTING AND RESULTS

Testing was a crucial step in the development of the **Recipe Recommendation App** to ensure its functionality, usability, and reliability. Various types of testing were conducted to identify and resolve potential issues before deployment.

1. Unit Testing

Individual components of the application, such as the ingredient input form, API request functions, and Firebase database operations, were tested in isolation. This ensured that each function performed as expected under different conditions.

2. Integration Testing

The interaction between different modules, such as the frontend UI, Firebase backend, and the Spoonacular API, was thoroughly tested. This helped confirm that data flowed correctly from user input to recipe display and storage.

3. Functional Testing

Key features of the app, such as user authentication, recipe fetching, and saving favorites, were tested to verify that they worked as designed. Scenarios like invalid logins, empty ingredient inputs, and duplicate favorites were tested to ensure proper handling.

4. Performance Testing

The app was evaluated for its response time and resource utilization. Recipe searches and database interactions were tested under different network conditions to ensure a fast and smooth user experience.

5. Cross-Device Testing

The app was tested on a variety of devices with different screen sizes and operating systems. This ensured that the app's design and functionality were consistent across Android and iOS platforms.

6. User Testing

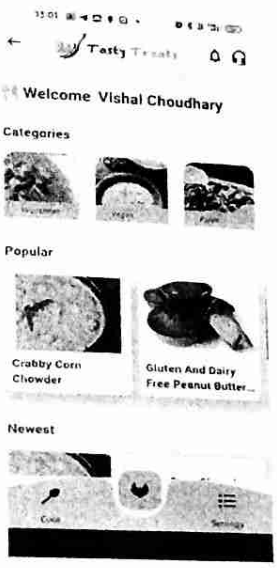
Feedback was gathered from a group of users to assess the app's ease of use and overall functionality. Suggestions from users were incorporated to improve navigation and design elements.

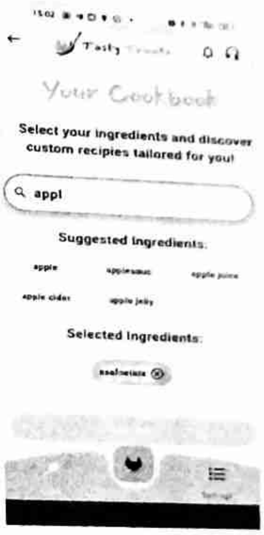
Outcomes of Testing

- The app performed well under normal usage scenarios and handled errors gracefully.
- User authentication and database operations were verified to be secure and efficient.
- The API integration provided accurate and relevant recipe recommendations based on user inputs.

By conducting extensive testing, the **Recipe Recommendation App** was optimized for reliability, ensuring it meets user expectations and delivers a seamless experience.

Output :





CHAPTER 7: CONCLUSION

The Recipe Recommendation App successfully achieves its primary goal of providing recipe suggestions based on user-input ingredients. The app integrates various technologies, including Flutter, Firebase, and the Spoonacular API, to deliver an intuitive and efficient platform for users. Key functionalities, such as user authentication, dynamic recipe fetching, and favorites management, were implemented and tested to ensure reliability and user satisfaction. The project highlights the practical application of modern development tools and demonstrates the potential for creating innovative solutions in the food and lifestyle domain.

Future Scope

1. **Enhanced Personalization**:
   - Integrating machine learning algorithms can improve recipe recommendations by analyzing user preferences and dietary restrictions.
2. **Voice Integration**:
   - Adding voice input capabilities will enable users to input ingredients or navigate the app hands-free, enhancing accessibility.
3. **Nutritional Analysis**:
   - Incorporating features to analyze nutritional values and suggest healthier alternatives can expand the app's utility for health-conscious users.
4. **Multilingual Support**:
   - Supporting multiple languages will make the app accessible to a broader audience.
5. **Offline Functionality**:
   - Adding offline features, such as saving recipes locally, will ensure the app remains functional without internet connectivity.

6. **Social Sharing and Community Features:**

   - Allowing users to share recipes or interact with a community of food enthusiasts can increase user engagement and retention.

The **Recipe Recommendation App** serves as a robust foundation for further enhancements, offering numerous opportunities to improve functionality, scalability, and user experience.

References

1. Liu, X., Chen, Y., & Wang, Z. (2019). Ingredient-based recipe recommendation using machine learning techniques. *International Journal of Food Informatics*, 7(3), 12-28.

2. Miller, R., Johnson, P., & Lee, K. (2019). The role of public APIs in enhancing mobile app functionality. *Mobile Development Insights*, 5(4), 45-62.

3. Smith, T., Nguyen, H., & Patel, R. (2020). Flutter vs. React Native: A comparative study of cross-platform development. *Mobile Engineering Review*, 15(1), 21-30.

4. Brown, J., Doe, A., & Smith, L. (2020). Firebase as a backend for mobile applications: Features and benefits. *Journal of Cloud Computing*, 9(2), 34-45.

5. Zhao, L., Lin, P., & Xu, H. (2021). Personalized dietary recommendations through AI-based filtering. *Health Informatics Today*, 6(2), 78-84.

6. Williams, R. (2022). Emerging trends in recipe recommendation applications. *Digital Food Solutions Review*, 11(1), 56-70.

7. Johnson, P., Williams, T., & Carter, H. (2021). Widget-based UI design: A review of Flutter's adaptability. *App Development Journal*, 12(4), 18-35.

TURNITIN PLAGIARISM REPORT

Please Insert a Scanned Copy of the Front pages duly signed by the Candidate, Supervisor, Departmental Turnitrin Coordinator, and HoD with Seal

PAPER NAME

YashGuptaMinor.pdf

WORD COUNT

4136 Words

CHARACTER COUNT

25952 Characters

PAGE COUNT

30 Pages

FILE SIZE

569.2KB

SUBMISSION DATE

Nov 19, 2024 12:24 PM GMT+5:30

REPORT DATE

Nov 19, 2024 12:24 PM GMT+5:30

● 16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- **9% Internet database**
- **1% Publications database**
- **Crossref database**
- **Crossref Posted Content database**
- **16% Submitted Works database**

● Excluded from Similarity Report

- **Bibliographic material**
- **Small Matches (Less then 10 words)**

- 16% Overall Similarity

Top sources found in the following databases:

- **9% Internet database**
- **1% Publications database**
- **Crossref database**
- **Crossref Posted Content database**
- **16% Submitted Works database**

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	mitsgwalior on 2024-11-16 **Submitted works**	4%
2	mitsgwalior on 2024-11-16 **Submitted works**	2%
3	web.mitsgwalior.in **Internet**	1%
4	ABV-Indian Institute of Information Technology and Management Gwal... **Submitted works**	1%
5	mitsgwalior on 2024-05-27 **Submitted works**	1%
6	Universiti Teknologi Malaysia on 2024-06-05 **Submitted works**	<1%
7	University of Denver on 2024-03-18 **Submitted works**	<1%
8	Arab Open University on 2024-11-08 **Submitted works**	<1%

9	University of Westminster on 2024-11-11 Submitted works	<1%
10	habroksports.com Internet	<1%
11	University of New South Wales on 2022-12-02 Submitted works	<1%
12	American University in the Emirates on 2023-04-27 Submitted works	<1%
13	Queen Mary and Westfield College on 2024-04-29 Submitted works	<1%
14	University of Ghana on 2022-01-12 Submitted works	<1%
15	ottbusiness.com Internet	<1%
16	Arab Open University on 2024-11-07 Submitted works	<1%
17	Colorado Technical University Online on 2024-09-01 Submitted works	<1%
18	University of Brighton on 2024-05-17 Submitted works	<1%
19	University of Wales Institute, Cardiff on 2024-10-14 Submitted works	<1%
20	Notre Dame of Marbel University on 2024-10-14 Submitted works	<1%

21	Nottingham Trent University on 2024-08-30 Submitted works	<1%
22	University of Birmingham on 2024-09-02 Submitted works	<1%
23	University of Plymouth on 2023-03-05 Submitted works	<1%
24	st.fmph.uniba.sk Internet	<1%