विश्वं शास्त्रे प्रतिष्ठितम्।

# MINI PROJECT REPORT

*On*

## SNAPRESCUE-AUTOMOBILE BREAKDOWN MANAGEMENT SYSTEM

*Submitted in partial fulfilment for the award of degree*

*Of*

***Master of Computer Applications***

*By*

**ROSEMARY B**

**(MLM24MCA-2047)**

Under the Guidance of

**Ms. DIVYA SB**

(HOD & Associate Professor, Department of Computer Applications)


**DEPARTMENT OF COMPUTER APPLICATIONS**

**MANGALAM COLLEGE OF ENGINEERING, ETTUMANOOR**

***(Affiliated to APJ Abdul Kalam Technological University)***

**OCTOBER 2025**

# MANGALAM COLLEGE OF ENGINEERING

**Accredited by NAAC& ISO 9001:2000 Certified Institution**

## DEPARTMENT OF COMPUTER APPLICATIONS

## VISION

To become a centre of excellence in computer applications,competent in the global ecosystem with technical knowledge,innovation with a sense of social commitment.

## MISSION

- To serve with state of the art education,foster advanced research and cultivate innovation in the field of computer applications.
- To prepare learners with knowledge skills and critical thinking to excel in the technological landscape and contribute positively to society.

## Program Educational Objectives

- PEO I :Graduates will possess a solid foundation and in-depth understanding of computer applications and will be equipped to analyze real-world problems, design and create innovative solutions, and effectively manage and maintain these solutions in their professional careers.
- PEO II: Graduates will acquire technological advancements through continued education, lifelong learning and research, thereby making meaningful contributions to the field of computing.
- PEO III: Graduates will cultivate team spirit, leadership, communication skills, ethics, and social values, enabling them to apply their understanding of the societal impacts of computer applications effectively.

## Program Specific Outcomes

- **PSO I**: Apply advanced technologies through innovations to enhance the efficiency of design development.
- **PSO II**: Apply the principles of computing to analyze, design and implement sustainable solutions for real world challenges.

# MAPPING OF PO-PSO-SDG

## 1. MAPPING WITH PROGRAM OUTCOMES (POs):-

| SL.NO | POs ADDRESSED | RELEVANCE TO PROJECT |
|---|---|---|
| 1 | PO1 | Applied computer science fundamentals (Node.js, MongoDB, Express) to build an intelligent roadside assistance system. |
| 2 | PO2 | Analyzed issues like delays, lack of real-time communication, and safety risks in breakdown management, leading to problem formulation. |
| 3 | PO3 | Designed a scalable web platform integrating geolocation, alerts, and dashboards to solve real-world breakdown challenges. |
| 4 | PO5 | Used modern IT tools (MongoDB, GPS API, Node.js, JavaScript) for real-time tracking, communication, and automation. |
| 5 | PO9 | Developed collaboratively, ensuring integration of different modules like user, workshop, and admin roles. |
| 6 | PO10 | Implemented role-based dashboards, notification systems, and status updates to ensure transparent communication. |
| 7 | PO11 | Applied software project management principles (Agile methodology) to deliver a cost-effective and scalable solution. |
| 8 | PO12 | Adopted latest web technologies and scalable architecture, adaptable for future upgrades like AI and IoT integration. |

**LIST OF PROGRAM OUTCOMES (POs):**

**PO1 – Engineering Knowledge** :Apply knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to solve complex engineering problems.

**PO2 – Problem Analysis**: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering

sciences.

**PO3 – Design/Development of Solutions**: Design solutions for complex engineering problems and design systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

**PO4 – Conduct Investigations of Complex Problems**: Use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data, and synthesis of information to provide valid conclusions.

**PO5– Modern Tool Usage** : Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling, to complex engineering activities with an understanding of the limitations.

**PO6 – The Engineer and Society**: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to professional engineering practice.

**PO7 – Environment and Sustainability**: Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of, and need for sustainable development.

**PO8 – Ethics** : Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

**PO9 – Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10 – Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11– Project Management and Finance**: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12 – Lifelong Learning**: Recognize the need for, and have the ability to engage in independent and life-long learning in the broadest context of technological change.

**2. MAPPING WITH PROGRAM SPECIFIC OUTCOMES (PSOs):**

| SL.NO | PSOs ADDRESSED | RELEVANCE TO PROJECT |
|---|---|---|
| 1 | PSO 1 | Used advanced full-stack technologies and geolocation services to develop an innovative roadside assistance system. |
| 2 | PSO 2 | Designed a sustainable digital roadside ecosystem reducing response delays, improving safety, and modernizing service delivery. |

## LIST OF PROGRAM SPECIFIC OUTCOMES (PSOs):

**PSO 1**: Apply advanced technologies through innovations to enhance the efficiency of design development.

**PSO 2**: Apply the principles of computing to analyze, design and implement sustainable solutions for real world challenges.

### 3. MAPPING WITH SUSTAINABLE DEVELOPMENT GOALS (SDGs):

| SDG NO | SDGs ADDRESSED | RELEVANCE TO PROJECT |
|---|---|---|
| SDG 8 | Decent Work & Economic Growth | Provides opportunities for workshops/service providers to gain more visibility and fair workload distribution. |
| SDG 9 | Industry, Innovation & Infrastructure | Promotes digital transformation in the automobile service sector through innovative IT-based roadside assistance. |
| SDG 11 | Sustainable Cities & Communities | Enhances urban mobility and emergency response efficiency, making cities safer and more resilient. |
| SDG 12 | Responsible Consumption & Production | Encourages efficient use of resources (time, workforce, and infrastructure) by optimizing workshop allocation. |
| SDG 17 | Partnerships for the Goals | Strengthens collaboration between users, workshops, and admin, creating a shared ecosystem for sustainable roadside assistance. |

## SUSTAINABLE DEVLOPMENT GOALS (SDGs):

**SDG 1 – No Poverty**-End poverty in all its forms everywhere.

**SDG 2 – Zero Hunger**-End hunger, achieve food security and improved nutrition, and promote sustainable agriculture.

**SDG 3 – Good Health and Well-Being**-Ensure healthy lives and promote well-being for all at all ages.

**SDG 4 – Quality Education**-Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.

**SDG 5 – Gender Equality**-Achieve gender equality and empower all women and girls.

**SDG 6 – Clean Water and Sanitation**-Ensure availability and sustainable management of water and sanitation for all.

**SDG 7 – Affordable and Clean Energy**-Ensure access to affordable, reliable, sustainable, and modern energy for all.

**SDG 8 – Decent Work and Economic Growth**-Promote sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all.

**SDG 9 – Industry, Innovation, and Infrastructure**-Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.

**SDG 10 – Reduced Inequality**-Reduce inequality within and among countries.

**SDG 11 – Sustainable Cities and Communities**-Make cities and human settlements inclusive, safe, resilient, and sustainable.

**SDG 12 – Responsible Consumption and Production**-Ensure sustainable consumption and production patterns.

**SDG 13 – Climate Action**-Take urgent action to combat climate change and its impacts.

**SDG 14 – Life Below Water**-Conserve and sustainably use the oceans, seas, and marine resources.

**SDG 15 – Life on Land** -Protect, restore, and promote sustainable use of terrestrial ecosystems, manage forests sustainably, combat desertification, halt and reverse land degradation, and halt biodiversity loss.

**SDG 16 – Peace, Justice, and Strong Institutions**- Promote peaceful and inclusive societies, provide access to justice for all, and build effective, accountable, and inclusive institutions.

**SDG 17 – Partnerships for the Goals** -Strengthen the means of implementation and revitalize the global partnership for sustainable development.

**MANGALAM COLLEGE OF ENGINEERING, ETTUMANOOR**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**OCTOBER 2025**


## ***DECLARATION***

*I hereby certify that the work which is being presented in the project entitled "**SNAPRESCUE**" submitted in the **DEPARTMENT OF COMPUTER APPLICATIONS** is an authentic record of my own work carried under the supervision of **DIVYA SB, HOD & ASSOCIATE PROFESSOR.** This study has not been submitted to any other institution or university for the award of any other degree. This report has been checked for plagiarism by the college and the similarity index is within permissible limits set by the college.*

*Name & Signature of Student*

*Date:*

*Place:*

**MANGALAM COLLEGE OF ENGINEERING, ETTUMANOOR**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**OCTOBER 2025**


## ***CERTIFICATE***

*This is to certify that the Project titled **"SNAPRESCUE"** is the bonafide record of the work done by **ROSEMARY B (MLM24MCA-2047)** of MCA in Computer Applications towards the partial fulfilment of the requirement for the award of MASTER OF COMPUTER APPLICATIONS **by APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**, during the academic year 2025-26.*

**Internal Examiner**

**Project Coordinator**

**Ms. Banu Sumayya s**

**Assistant Professor**

**Department Of Computer Applications**

**Project Guide**

**Ms. Divya SB**

**HOD & Associate Professor**

**Department Of Computer Applications**

**Head of the Department**

**Ms. Divya SB**

**Associate Professor**

**Department of Computer Applications**

# ACKNOWLEDGEMENT

I am greatly indebted to the authorities of Mangalam College of Engineering for providing the necessary facilities to successfully complete my Project on the topic "SnapRescue-Automobile Breakdown Management Application".

I express my sincere thanks to **Dr. Vinodh P Vijayan**, Principal, Mangalam College of Engineering for providing the facilities to complete my Project successfully.

I thank and express my solicit gratitude to **Ms. Divya SB,** HOD, Department of Computer Applications, Mangalam College of Engineering, for her invaluable help and support which helped me a lot in successfully completing this Project work.

I express my gratitude to my Internal Guide, **Ms. Divya SB**, Associate professor, Department of Computer Applications for the suggestions and encouragement which helped in the successful completion of our Project.

Furthermore, I would like to acknowledge with much appreciation the crucial role of the faculties especially Project coordinator, **Ms. Banu Sumayya S**, Department of Computer Applications, Mangalam College of Engineering, who gave the permission to use all the required equipment and the necessary resources to complete the presentation & report.

Finally, I would like to express my heartfelt thanks to my parents who were very supportive both financially and mentally and for their encouragement to achieve my goal.

**ROSEMARY B**
**(MLM24MCA-2047)**

# ABSTRACT

SnapRescue is an innovative web-based platform designed to provide instant roadside assistance for vehicle breakdowns. The system connects stranded vehicle owners with nearby workshops and service providers in real time, ensuring quick and reliable rescue operations. Built using Node.js, Express, MongoDB, HTML, CSS, and JavaScript, SnapRescue integrates both frontend and backend technologies to deliver a seamless user experience.

The application features a role-based system comprising users, workshops, and administrators. Users can register, log in, and report vehicle breakdowns by entering essential details such as location and problem description. Nearby workshops receive immediate alerts about new breakdown requests and can accept or decline them through their dashboards. Administrators oversee the platform's activities, managing users, workshops, and overall operations.

By leveraging real-time notifications, secure authentication, and a user-friendly interface, SnapRescue minimizes response time during emergencies and enhances the efficiency of roadside assistance services. This project demonstrates the potential of technology in improving safety, convenience, and service delivery for vehicle owners in distress.

***Keywords:***

***Breakdown Management, Role-Based Access Control, Leaflet.js Map Integration, Online Breakdown Request Portal, Emergency Response, Real-time Tracking.***

.

| **Mapping with Sustainable Development Goals** | **Industry, Innovation and Infrastructure** |
|---|---|
| Decent Work and Economic Growth | Provides opportunities for local workshops and service providers to gain more visibility, ensuring fair workload distribution and contributing to economic empowerment |
| Industry, Innovation and Infrastructure | Promotes digital transformation in the automobile service sector by leveraging innovative web and mobile-based roadside assistance technology for improved service delivery. |

| | |
|---|---|
| Sustainable Cities & Communities | Enhances urban mobility and emergency response systems by ensuring timely roadside assistance, thereby making cities safer, more efficient, and more resilient. |
| Responsible Consumption & Production | Encourages efficient utilization of resources — such as time, manpower, and vehicle infrastructure — through optimized allocation of nearby workshops and service providers. |
| Partnerships for the Goals | Strengthens collaboration between users, workshops, and administrators by creating an integrated digital ecosystem that supports sustainability and shared growth. |

# LIST OF FIGURES

## List of Abbreviations

| ABBREVIATION | FULL FORM |
|---|---|
| **DFD** | Data Flow Diagram |
| **UML** | Unified Modeling Language |
| **AI** | Artificial Intelligence |
| **CRUD** | Create, Read, Update, Delete |
| **DBMS** | Database Management System |
| **ER** | Entity Relationship |
| **HTML** | HyperText Markup Language |
| **HTTP** | HyperText Transfer Protocol |
| **JSON** | JavaScript Object Notation |
| **PDF** | Portable Document Format |
| **REST** | Representational State Transfer |
| **UAT** | User Acceptance Testing |
| **UI** | User Interface |
| **UX** | User Experience |
| **API** | Application Programming Language |
| **CSS** | Cascading Style Sheets |

| | |
|---|---|
| **JSON** | JavaScript Object Notation |
| **URL** | Uniform Resource Locator |

# CHAPTER-I

# INDRODUCTION

## 1.1 BACKGROUND

Vehicle breakdowns are an unavoidable aspect of road transportation, often leading to delays, safety concerns, and inconvenience for both drivers and service providers. With the growing number of vehicles on the road, the demand for an efficient, reliable, and technology-driven roadside assistance system has become increasingly important. Conventional breakdown assistance methods primarily rely on manual communication through phone calls or roadside services, which can result in delayed responses, inaccurate location sharing, and inefficient coordination between drivers and workshops. These challenges highlight the need for an intelligent system capable of automating and optimizing the breakdown response process.

In response to these issues, the Snap Rescue project aims to provide an integrated digital platform that connects users, workshops, and administrators in real time. The system utilizes advanced web technologies such as Node.js, Express, Mongo DB, and JavaScript to create a seamless environment for managing roadside emergencies. By incorporating geolocation tracking, automated notifications, and role-based dashboards, Snap Rescue ensures rapid communication between stranded users and nearby workshops. This not only minimizes response time but also enhances transparency and coordination among all stakeholders involved in the rescue process.

Traditional systems for roadside assistance often suffer from poor accessibility and lack of automation. Snap Rescue addresses these shortcomings by introducing a structured architecture that supports three distinct user roles—user, workshop, and admin—each with specific functionalities. Users can register and report breakdown incidents with accurate location details, workshops receive immediate alerts and can respond to requests promptly, while administrators monitor overall activity and ensure service reliability. Through this automated workflow, the system significantly improves efficiency, accountability, and user satisfaction.

Furthermore, the integration of real-time data and intelligent web services enables Snap Rescue to move beyond conventional methods by providing immediate, location-based assistance and digital record-keeping. The project's emphasis on speed, accuracy, and automation underscores its

potential to transform the way roadside emergencies are managed. By leveraging modern web technologies and cloud databases, Snap Rescue aspires to create a smart, scalable, and dependable roadside assistance ecosystem that ensures timely support for users, enhances operational efficiency for workshops, and contributes to safer and more reliable road transportation infrastructure**.**

## 1.2 INDRODUCTION

The increasing number of vehicles on the road and the growing complexity of transportation networks have led to a significant rise in roadside breakdown incidents, emphasizing the urgent need for intelligent, efficient, and real-time assistance systems. Traditional methods of handling vehicle breakdowns rely heavily on manual communication between drivers and roadside service providers, often resulting in delays, miscommunication, and limited situational awareness. These limitations not only prolong the response time but also pose potential safety risks to stranded individuals, particularly in remote or high-traffic areas. Addressing these challenges requires a paradigm shift from conventional manual operations to automated, data-driven systems capable of providing immediate and location-aware support.

In response to this pressing need, the proposed Snap Rescue project introduces an innovative web-based framework that seamlessly integrates real-time geo location services with dynamic data management to revolutionize the breakdown assistance process. The system leverages modern web technologies Node.js**,** Express**,** Mongo DB, and JavaScript to create a robust and interactive platform that connects vehicle owners with nearby workshops and administrative authorities. By automating key processes such as user registration, authentication, breakdown reporting, and workshop allocation, Snap Rescue ensures swift, transparent, and reliable roadside assistance.

The proposed system operates through four principal modules User Interface, Data Processing, Notification Management, and Administrative Control each contributing to an intelligent workflow designed to optimize user experience and service efficiency. The integration of real-time location tracking and automated notification systems enables the nearest available workshop to respond promptly to user requests, minimizing downtime and improving safety.

Furthermore, the project's role-based access architecture ensures secure data handling, with distinct functionalities for users, workshops, and administrators.

Notably, Snap Rescue emphasizes the importance of automation and intelligent coordination, moving beyond traditional call-based assistance models. The system's scalability allows for future enhancements, including AI-powered fault prediction**,** route optimization, and mobile integration**,** paving the way toward a fully smart roadside ecosystem. The project's structured framework ensures systematic data handling, operational transparency, and a seamless user experience, ultimately contributing to the modernization of emergency vehicle support systems.

Ultimately, Snap Rescue aims to redefine roadside assistance by making it faster, smarter, and more accessible. Through the integration of real-time communication, secure data management, and automated coordination, the system aspires to reduce the response time during vehicle breakdowns, enhance road safety, and ensure reliable support for every driver in need. By leveraging the power of modern web technologies, Snap Rescue represents a significant step toward building a connected, efficient, and resilient transportation assistance infrastructure for the future

## 1.3 PROBLEM STATEMENT

In the modern era of rapid technological advancement and increasing vehicular dependency, road transportation has become an integral part of daily life. However, with the rising number of vehicles on the road, the frequency of roadside breakdowns and emergency situations has also escalated, posing significant challenges to drivers and service providers alike. Traditional breakdown assistance methods rely predominantly on manual communication through help lines or roadside services, which are often hindered by delayed responses, lack of accurate location tracking, and inefficient coordination between drivers and workshops. These limitations not only prolong vehicle downtime but also endanger the safety of stranded individuals, particularly in isolated or high-traffic areas.

This persistent gap in real-time communication and response efficiency forms the central problem that the Snap Rescue project seeks to address. The absence of an integrated, automated

platform for connecting users with nearby workshops during vehicle emergencies highlights the need for a more intelligent, data-driven, and responsive solution. By leveraging modern web technologies and geo location services, Snap Rescue aims to overcome these challenges through an interactive, real-time system that streamlines communication, minimizes response time, and enhances overall operational efficiency.

The project envisions the creation of a robust, scalable, and secure platform that not only digitizes the roadside assistance process but also ensures reliability, transparency, and accessibility for all users. By introducing automation and intelligent coordination, Snap Rescue aspires to transform the traditional model of breakdown management into a smart, connected, and proactive system capable of redefining roadside assistance for the modern transportation ecosystem.

## 1.4 MOTIVATION

The motivation behind developing the Snap Rescue system arises from the pressing need to enhance road safety, efficiency, and accessibility in emergency vehicle breakdown scenarios. In an era of increasing vehicular movement and reliance on road transport, drivers frequently encounter unexpected mechanical failures or accidents that leave them stranded in unfamiliar or unsafe locations. Existing roadside assistance mechanisms often depend on manual communication through phone calls or third-party services, which are limited by response delays, lack of precise location data, and ineffective coordination between users and service providers. These challenges not only inconvenience vehicle owners but also pose potential safety hazards, particularly during nighttime travel or in remote areas.

The development of Snap Rescue **i**s inspired by the vision of transforming the conventional roadside assistance process into a digital, real-time, and intelligent system. The motivation stems from the desire to bridge the technological gap between distressed drivers and nearby workshops or service stations. By leveraging the capabilities of modern web technologies, geo location services, and database integration, Snap Rescue aims to create a seamless platform where users can instantly report breakdown incidents and receive timely help without the need for extensive manual intervention.

Furthermore, the project is motivated by the broader objective of promoting *smart mobility* and *user-centric innovation* in the transportation domain. The integration of automated alerts, live tracking, and intelligent coordination ensures that assistance is provided efficiently, minimizing waiting time and reducing the potential for secondary road hazards. In addition, the platform seeks to empower workshops and service providers by offering a structured interface for managing requests, improving their operational visibility, and fostering trust through transparency and accountability.

At its core, the motivation for developing Snap Rescue lies in the ambition to redefine roadside assistance from a reactive service into a proactive, technology-driven ecosystem. By utilizing real-time data flow and intelligent matching algorithms, the system aspires to make vehicle rescue faster, safer, and more reliable. Ultimately, the project aims to contribute to a future where technology ensures not just convenience but also the safety, confidence, and well-being of every traveler on the road.

## 1.5 SCOPE

The scope of the Snap Rescue project encompasses the design, development, and deployment of a comprehensive web-based platform aimed at providing immediate roadside assistance to vehicle owners during breakdown or emergency situations. The system is conceptualized to streamline the interaction between stranded users and nearby workshops, ensuring efficient communication, accurate location tracking, and prompt response. It focuses on creating a user-friendly interface that facilitates quick reporting of breakdown incidents, automatic location detection, and real-time updates on the status of the assistance request.

Within the functional boundaries of this project**,** Snap Rescue integrates multiple modules, including user authentication**,** role-based dashboards (for users, workshops, and administrators), and incident managemen**t** features. Users can register or log in to report breakdowns, while workshops can view and respond to nearby assistance requests. The administrator oversees overall operations, ensuring data integrity, managing users, and monitoring system performance. Additionally, the system incorporates real-time alerts**,**

notification mechanisms, and database-driven storage using Mongo DB, ensuring data consistency and scalability.

The technological scope of the project extends to the use of Node.js**,** Express.js**,** Mongo DB, HTML**,** CSS**,** and JavaScript**,** providing a full-stack environment capable of handling both client-side and server-side functionalities. The integration of geo location services enhances the precision of identifying the user's location, while future scalability provisions allow for integration with mobile applications, IoT sensors, and AI-driven predictive maintenance systems.

From a societal and practical standpoint, the project's scope also includes improving the overall roadside assistance ecosystem by minimizing response time, enhancing driver safety, and fostering digital transformation among local workshops. By offering a structured, data-driven, and accessible service platform, Snap Rescue aspires to serve both individual users and professional service providers, creating a mutually beneficial network that prioritizes efficiency, safety, and user satisfaction. Ultimately, the scope of Snap Rescue extends beyond merely addressing immediate breakdown situations—it envisions a smarter, more connected, and responsive roadside assistance framework that can evolve alongside future advancements in intelligent transportation and vehicular technologies.

# CHAPTER-2
# LITERATURE REVIEW

## 2.1 Vehicle Breakdown Assistance System [Dr. A. Vinoth, Mr. V. Maheshwaran (2025)]"International journal of Research Publications and Reviews"

The paper titled "Vehicle Breakdown Assistance System" presents an effective approach to providing timely roadside help to drivers experiencing vehicle breakdowns through web-based and location-based technologies. This research focuses on improving driver safety, reducing response time during breakdown situations, and enhancing the overall efficiency of emergency assistance systems.

Key Aspects:

• Problem Identification: The study identifies common challenges faced by drivers during unexpected vehicle breakdowns, such as difficulty finding nearby workshops, delayed assistance, and lack of real-time communication. The system addresses these issues by integrating GPS tracking and automated notifications.

• System Design**:** The proposed system features a web and mobile-based platform connecting users, workshops, and administrators. Users can report a vehicle breakdown, and their real-time location is captured to notify the nearest available workshop. Workshops can accept requests and dispatch assistance promptly.

• Technology Used**:** Modern web technologies such as Node.js, Express.js, MongoDB, and Google Maps API are employed for location tracking and data management. These ensure fast communication, secure data handling, and efficient coordination between users and service providers.

• Modules and Functionality**:** The system includes modules for user authentication, breakdown request management, workshop dashboard, and admin control panel. Each module ensures transparency, effective task allocation, and monitoring of real-time assistance.

• Advantages: The system provides a user-friendly and reliable solution, removing the traditional need for manual phone calls or searching for workshops. It delivers assistance based on proximity and availability, saving time and improving driver safety.

• Impact and Innovation**:** The model emphasizes real-time connectivity between users and workshops, significantly reducing waiting periods for roadside assistance. It represents a step toward digitizing emergency vehicle support and aligns with smart city and intelligent transport system initiatives.

**2.2 On-Road Vehicle Breakdown Assistance System [Ms. S. Gowthami, Athira Nagarajan, C. Dharsha, K. Dharshini Priya, R. Gowri (2024)]"International journal of Research and Analytical Reviews"**

The paper titled "On-Road Vehicle Breakdown Assistance System" presents an innovative approach to providing immediate roadside help to drivers experiencing vehicle breakdowns using web-based and location-based technologies. This research focuses on improving driver safety, minimizing response time during breakdown situations, and enhancing the overall efficiency of emergency assistance systems.

Key Aspects:

• Problem Identification: The study begins by identifying the frequent challenges faced by drivers during unexpected vehicle breakdowns, such as the lack of immediate help, difficulty locating nearby workshops, and communication delays. The system aims to address these issues by integrating digital communication and GPS-based tracking.

• System Design: The proposed system incorporates a web-based platform that connects users, workshops, and administrators. The user can report a vehicle breakdown through the system, which captures their location in real time and notifies the nearest available workshop. The workshop, in turn, can accept the request and dispatch assistance promptly.

• Technology Used: The paper highlights the use of modern web development tools and technologies such as Node.js, Express.js, MongoDB, and Google Maps API for location tracking.

These technologies ensure fast data transmission, secure communication, and efficient coordination between users and service providers.

• Modules and Functionality: The system includes key modules such as user authentication, breakdown request management, workshop dashboard, and admin control panel. Each module works together to ensure transparency, efficient task allocation, and monitoring of real-time assistance.

• Advantages: This system provides a user-friendly and reliable solution that eliminates the traditional need for manual phone calls or searching for service centers. It ensures that users receive help based on proximity, availability, and service type, thus saving time and enhancing safety.

• Impact and Innovation: The proposed model emphasizes real-time connectivity between users and workshops, reducing the waiting period for roadside assistance. It represents a significant step toward digitizing roadside emergency management and supports smart city and intelligent transport system initiatives.

**2.3 Vehicle Breakdown Assistance Management System [Prof. Poorva Wagh, Dipesh Bais, Md. Arzaan Sheikh (2024)]"International Journal of Computer Applications"**

The paper titled "Vehicle Breakdown Assistance Management System" presents an advanced digital solution aimed at simplifying and modernizing the process of providing roadside assistance during vehicle breakdowns. The study emphasizes the integration of web-based systems and GPS technology to connect stranded drivers with nearby service providers in real time.

Key Aspects:

• Objective: The paper focuses on minimizing the delay and inconvenience experienced by drivers during vehicle breakdowns by creating a centralized system that efficiently manages communication between users, workshops, and administrators.

• System Overview: The proposed model automates the traditional manual process of finding help by allowing users to report breakdowns through a web or mobile interface. Once a request is

submitted, the system uses location tracking to identify nearby mechanics or workshops and notifies them for prompt response.

• Technologies Used: The system is developed using modern technologies such as HTML, CSS, JavaScript, Node.js, and Mongo DB, with GPS and Google Maps API integration. These technologies work together to ensure real-time location updates, efficient routing, and effective communication.

• Modules: The system comprises three major modules—User, Workshop, and Admin. The user module allows reporting of breakdowns and tracking assistance, the workshop module manages received requests, and the admin module oversees operations and ensures smooth coordination between all entities.

• Benefits: The system reduces response time, enhances communication, and ensures timely support. It provides transparency, real-time tracking, and an improved user experience. It also supports digital record-keeping for future reference and data analysis.

• Future Implications: The paper highlights that with further enhancement, such systems can be expanded to include automated diagnostics, AI-based fault detection, and integration with smart transportation networks to support large-scale deployment in urban areas.

### 2.4 Implementation of an Automobile Breakdown Service Provider (ABSP) Model [Oluwatofunmi O. Adetunji, Inioluwa A. Alake, Oluwabukola F. Ajayi, and Yaw A. Mensah (2023)]"International Journal of Scientific and Research Publications"

The research paper titled "Implementation of an Automobile Breakdown Service Provider (ABSP) Model" focuses on designing an automated and efficient system that provides quick roadside assistance to vehicle owners experiencing mechanical failures. The study emphasizes the use of a centralized digital platform to connect users, service providers, and administrators in real time, thereby minimizing downtime and improving the overall user experience.

Key Aspects:

• Objective: The main goal of the study is to develop a smart system that ensures timely assistance to stranded drivers by integrating digital service request handling, location tracking, and automatic workshop allocation.

• System Design: The ABSP model is structured to enable seamless interaction between vehicle owners and nearby mechanics. It uses a cloud-based database to store user and vehicle data, which allows for efficient management of breakdown requests and service responses.

• Technologies Used: The implementation makes use of web technologies such as PHP, MySQL, and JavaScript for backend and frontend operations. GPS integration and mobile connectivity ensure that users can easily locate and contact nearby service providers in emergency situations.

• Functionality: Users can register, log in, and raise breakdown requests through a user-friendly interface. The system then automatically matches the request with the nearest available service provider and provides real-time updates on the service status.

• Benefits: The ABSP model enhances convenience, reduces waiting time, and ensures safety for users stranded in remote or unsafe locations. It promotes automation, transparency, and faster communication between all entities involved.

• Future Prospects: The paper suggests incorporating artificial intelligence and predictive analytics to forecast vehicle failures, along with integrating payment gateways and expanding the system's coverage to rural areas for improved accessibility.

**2.5 Onroad Fuel and Breakdown Management [Shalima S (2023)]"International Journal of Emerging Technologies and Advanced Engineering"**

The paper titled "Onroad Fuel and Breakdown Management" focuses on developing a comprehensive system that provides real-time assistance for both vehicle breakdowns and fuel emergencies. The research highlights the importance of integrating technology to create a more reliable, efficient, and user-friendly roadside assistance platform for drivers facing sudden mechanical failures or fuel shortages during travel.

Key Aspects:

• Objective: The main aim of the system is to provide on-demand support for vehicle breakdowns and fuel delivery by connecting drivers with nearby workshops or service providers through a centralized application.

• System Overview: The system is designed to allow users to request help instantly through a web or mobile interface. Upon request, the system automatically locates the nearest service provider using GPS and dispatches the appropriate assistance, whether it be towing, repair, or fuel delivery.

• Technology Stack: The implementation uses web-based technologies along with GPS and Google Maps API for live location tracking. A backend database maintains user profiles, service history, and real-time updates to ensure accurate and efficient service delivery.

• Functionality: Users can register and log in to the platform, report their location and problem type, and receive confirmation once a service provider is assigned. The system ensures that help reaches the user in minimal time by optimizing location and route data.

• Advantages: This model enhances driver safety, reduces response time during emergencies, and ensures that even in remote areas, users can access assistance promptly. It also minimizes stress and uncertainty for travelers experiencing breakdowns or fuel shortages.

• Future Scope: The paper proposes integrating AI-based predictive maintenance, automated service dispatching, and digital payment systems in future versions to improve accuracy, speed, and convenience for users.

**2.6 Vehicle Breakdown Management System [Dr. Kousalya Devi, Manikandan S, Naveen Kumar R (2024)]"International Journal of Research publications and Reviews"**

The paper titled "Vehicle Breakdown Management System" presents an intelligent and automated solution designed to assist vehicle owners in emergency breakdown situations. The study emphasizes developing a system that ensures immediate help through real-time communication between users, workshops, and towing services.

Key Aspects:

• Objective: The primary goal of the system is to reduce the time taken for assistance during vehicle breakdowns and to provide a reliable platform that connects users to nearby service providers efficiently.

• System Overview: The proposed system incorporates user registration, location tracking, and service allocation functionalities. When a breakdown occurs, users can log into the platform, enter details of their issue, and request help. The system uses GPS to locate the user and automatically assigns the nearest available workshop or service provider.

• Technology Stack: The system is built using modern web technologies combined with GPS and Google Maps API integration for precise location tracking. A centralized database is used to maintain records of users, workshops, and service requests, ensuring smooth data flow and efficient management.

• Functionality: Users can report the type of breakdown—mechanical failure, flat tire, or towing need—and the system instantly notifies the respective workshop. Service providers can accept or reject requests, ensuring a quick and organized workflow.

• Advantages: This approach minimizes response time and enhances driver safety. It eliminates the manual process of searching for nearby help, thereby improving the overall efficiency and convenience of roadside assistance.

• Future Scope: The paper suggests integrating AI-based predictive maintenance and mobile app notifications to alert users before potential breakdowns occur. Additionally, automated payment options and multilingual support could be included to make the system more user-friendly and widely accessible.

**2.7 On Road Vehicle Breakdown Help Assistance Web Application [C. S. Aravinthan, M. Hemalatha(2024)]"International Journal of Scientific Research in Science and Technology"**
The paper titled "On Road Vehicle Breakdown Help Assistance Web Application" presents a web-based platform designed to provide timely roadside assistance to vehicle owners. It focuses on

improving response time, connecting users with nearby certified mechanics, and ensuring safe and efficient communication during emergencies.

Key aspects:

• Problem Identification: The study highlights delays in reaching stranded vehicles, lack of a centralized communication platform, and difficulty locating nearby workshops as key challenges.
• System Design**:** The system provides a web interface where users can report breakdowns, share real-time location, and connect instantly with nearby workshops.
• Technology Used**:** Utilizes web technologies for frontend/backend development, cloud databases for storing user requests, and real-time notifications for service updates.
• Modules and Functionality**:** Includes modules for user registration, breakdown reporting, workshop management, and notifications.

• Advantages**:** Reduces waiting time for users, ensures quick assistance, and provides a reliable method to find nearby workshops.

• Impact and Innovation**:** Emphasizes real-time connectivity, improving efficiency and safety for drivers experiencing breakdowns.

**2.8 Review of Challenges and Solutions in Web Based Vehicle Breakdown Assistance System [R. Sharma, D. Verma, D. Singh, H. Sharma, P. Nishad (2022)]"International Journal for Research In Engineering"**

The paper titled "Review of Challenges and Solutions in Web Based Vehicle Breakdown Assistance System" discusses common issues faced by roadside assistance platforms and proposes strategies to overcome them.

Key Aspects:

• Problem Identification**:** Identifies delays, difficulty locating nearby workshops, and lack of system reliability as major issues.

• System Design**:** Recommends centralized web-based systems to manage user requests efficiently.
• Technology Used**:** Emphasizes using modern web technologies, GPS tracking, and real-time notifications.
• Modules and Functionality**:** Focuses on user request management, workshop assignment, and response tracking.

• Advantages: Enhances service reliability, reduces response time, and ensures proper coordination.
• Impact and Innovation: Provides guidelines for improving web-based vehicle breakdown assistance platforms.

**2.9 On-Road Vehicle Breakdown Assistance System [Ms. S. Gowthami, Athira Nagarajan, C. Dharsha, K. Dharshini Priya, R. Gowri (2024)]"International Journal of Engineering and Technology"**

The paper titled *"On-Road Vehicle Breakdown Assistance System"* presents an innovative approach to providing immediate roadside help to drivers experiencing vehicle breakdowns using web-based and location-based technologies. This research focuses on improving driver safety, minimizing response time during breakdown situations, and enhancing the overall efficiency of emergency assistance systems.

Key Aspects:

- Problem Identification: The authors highlight common issues faced by drivers during breakdowns, including the lack of immediate help, difficulty in locating nearby service centers, and delays in communication. The proposed system aims to tackle these issues through digital communication tools and real-time GPS tracking.
- System Design: A web-based platform connects users, workshops, and administrators. When a user reports a breakdown, the system captures their live location and notifies the nearest workshop. The workshop can accept the request and dispatch help accordingly.
- Technology Used: The system is developed using modern web technologies such as Node.js, Express.js, MongoDB, and the Google Maps API for accurate location tracking and efficient service coordination.

- Modules and Functionality: It includes several functional modules such as user authentication, breakdown request submission, workshop dashboard for managing requests, and an admin panel for overall system monitoring.
- Advantages: The system offers a user-friendly interface and eliminates the need for manual calls or physical search for service centers. It prioritizes service based on location proximity and availability, thereby saving time and enhancing user safety.
- Impact and Innovation: The proposed system strengthens the digital infrastructure for roadside assistance. It supports smart transportation goals by ensuring quick connectivity and reliable support during emergencies.

**2.10 On-Road Vehicle Breakdown Assistance System [Prof. Poorva Wagh, Dipesh Bais, Md. Arzaan Sheikh (2024)]"International Journal for Research in Applied Science And Engineering Technology"**

The paper titled *"Vehicle Breakdown Assistance Management System"* proposes an advanced digital platform to efficiently manage vehicle breakdown situations. The system focuses on improving coordination between vehicle users and nearby service providers through automated processes and centralized management.

Key Aspects:

- Objective: The primary goal of the system is to reduce the delay in roadside assistance by creating an organized and automated breakdown management system.
- System Overview: Users can report vehicle breakdowns via the system interface. The platform identifies the user's location and assigns the nearest available service provider, who can accept and respond to the request.
- Technical Implementation: While specific technologies are not detailed, the system likely incorporates standard web technologies and mapping tools to facilitate real-time tracking and communication between users and service providers.
- Modules: The system may consist of user registration, request tracking, service provider interface, and an admin control panel for monitoring and reporting.

- Benefits: It ensures faster service delivery, accurate tracking of breakdown locations, and better service management. It minimizes manual efforts and enhances efficiency in emergency handling.
- Significance: This system contributes to the modernization of roadside emergency services and promotes a tech-driven approach to managing vehicle breakdowns, aligning with digital transformation in transport services.

# CHAPTER-3

# PROPOSED SYSTEM

## 3.1 Users

Users interact with the Snap Rescue web application by performing essential actions such as registration, login, and reporting breakdown incidents. The system ensures a user-friendly interface, allowing individuals to request emergency roadside assistance quickly and efficiently**.**

## 3.2 Registration

During registration, users provide necessary information such as name, contact details, vehicle information, and password credentials. This data is validated for accuracy and securely stored in the Mongo DB database. The registration process ensures that only legitimate users can access the system, providing authentication for future logins.

## 3.3 Login

Once registered, users can log in using their credentials. The authentication module, developed using Node.js and Express sessions, verifies credentials by matching them with the stored data in the database. This step ensures secure access and prevents unauthorized users from entering the system.

## 3.4 Breakdown Reporting

After logging in, users can report vehicle breakdowns by entering details such as vehicle type, problem description, and current location. The system integrates **geo location tracking** to automatically capture the user's coordinates, allowing the nearest workshop to be identified efficiently.

## 3.5 Data Processing and Alert Generation

Upon submission of a breakdown report, the system processes the input data and triggers an automated alert to nearby workshops. The alert includes key information such as the user's location, breakdown type, and contact details. This step reduces manual intervention and accelerates response time.

## 3.6 Workshop Response

Workshops receive notifications of incoming breakdown alerts through their dedicated dashboards. They can view, accept, or decline the request based on availability. Upon acceptance, the workshop's details are shared with the user, ensuring quick and transparent communication between both parties.

## 3.7 Notification and Communication System

The system includes automated notifications that inform workshops of new requests and users of status changes. These notifications may be sent via in-app messages or emails. The communication framework ensures that both users and workshops stay updated throughout the process.

## 3.8 Database Management

The Snap Rescue database, developed using Mongo DB, stores structured data for users, workshops, and breakdowns. Each entry is linked via unique identifiers, allowing for efficient data retrieval, scalability, and system stability. Regular backups and validation ensure data consistency and reliability

## 3.9sa Security and Authentication

User data and workshop information are protected through encrypted passwords and secure session handling. The system uses **bcrypt** for password hashing and **Express-session** for secure session management. Only authorized users can access or modify information based on their assigned roles.

# CHAPTER-4

# METHODOLOGY.

The methodology adopted for the Snap Rescue project is a systematic and structured approach designed to ensure efficient development, seamless functionality, and user-centered performance. The project follows a modular design framework, dividing the system into distinct yet interconnected components that collectively deliver a cohesive and responsive roadside assistance solution. Each stage of the methodology focuses on achieving scalability, security, and reliability while maintaining simplicity for end users.

The project employs the Agile Development Model, which emphasizes iterative progress, continuous feedback, and adaptive refinement. This methodology allows for flexible planning, progressive development, and early identification of potential issues during implementation. The major phases of the project methodology are as follows:

## Phase 1: Requirement Analysis

Implementation This initial phase involves identifying and defining the functional and non-functional requirements of the Snap Rescue system. A detailed study was conducted to understand the pain points faced by drivers during vehicle breakdowns and the challenges encountered by workshops in responding efficiently. The key requirements identified include:

- A user-friendly interface for quick breakdown reporting.
- Integration of real-time location tracking.
- A secure authentication system for users and workshops.
- A centralized database for managing requests and responses.
  The outcome of this phase provides a clear blueprint for subsequent design

## Phase 2: System Design

In this phase, the overall system architecture and module design are developed. The project adopts a three-tier architecture consisting of:

- Frontend Layer – Responsible for user interaction through web interfaces built with HTML, CSS, and JavaScript.
- Backend Layer – Managed using Node.js and Express.js, handling routing, authentication, and server-side logic.
- Database Layer – Implemented using Mongo DB, ensuring structured data storage, retrieval, and management. The design phase also includes creating data flow diagrams (DFD), entity-relationship diagrams (ERD), and workflow charts to visualize the process of user requests, workshop responses, and administrative monitoring.

## Phase 3: Implementation

During this stage, each module is coded and integrated to form the complete system. The primary modules implemented are:

- User Module: Enables registration, login, and breakdown request submissions.
- Workshop Module: Allows workshops to manage and respond to nearby breakdown alerts.
- Admin Module: Provides oversight of system activity, user management, and data analytics.

  The backend logic handles authentication, role-based access, and data communication between the server and the database. Frontend pages are designed for clarity and responsiveness, ensuring seamless navigation.

## Phase 4: Integration and Testing

Once individual modules are developed, they are integrated into a unified system and subjected to rigorous testing. The testing methodologies employed include:

- Unit Testing – Verifying the correctness of each individual module.
- Integration Testing – Ensuring seamless data flow between modules.
- System Testing – Validating the entire platform against functional requirements.
- User Acceptance Testing (UAT) – Gathering feedback from potential users to enhance usability and reliability. Defects identified during testing are rectified to ensure system stability and robustness.

## Phase 5: Deployment

After successful testing, the Snap Rescue platform is deployed on a secure web server. The deployment phase involves configuring the backend server, connecting the Mongo DB database, and ensuring that all APIs are functional. The live environment allows real-time interactions between users, workshops, and administrators.

The adopted methodology ensures that Snap Rescue evolves as a reliable, scalable, and efficient roadside assistance system. By combining modern web technologies with systematic software engineering principles, the methodology establishes a strong foundation for future innovation and expansion of intelligent transport support systems.

# CHAPTER-5

# SYSTEM ARCHITECTURE

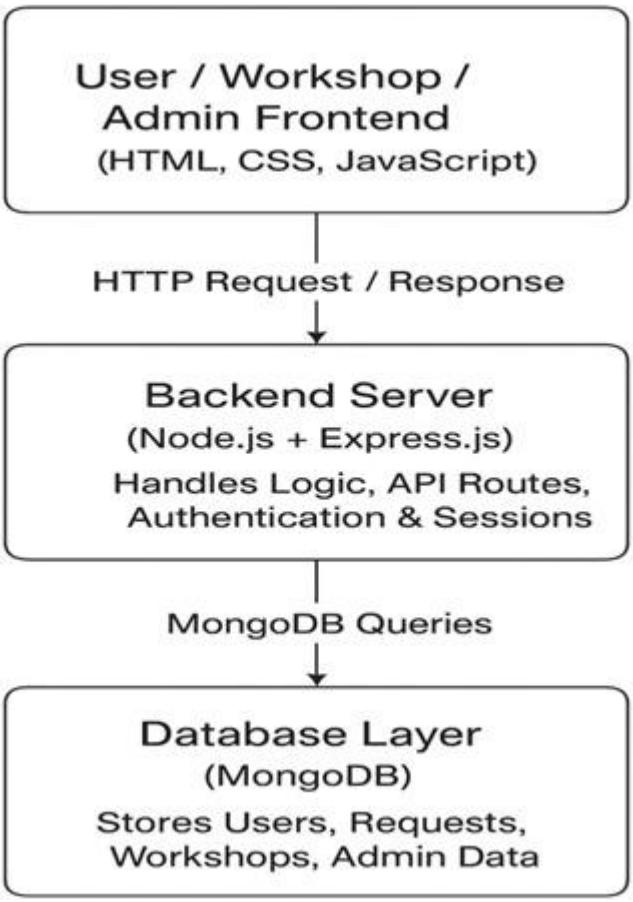


**Figure 5.1 System architecture**

The **system architecture** of the *Snap Rescue* project is designed as a **three-tier architecture**, ensuring scalability, efficiency, and seamless interaction between users, workshops, and administrators. It integrates both **frontend and backend technologies** with a robust database layer to deliver real-time roadside assistance and intelligent service management. The architecture ensures that every process—from user authentication to breakdown resolution—is handled securely and efficiently.

At its core, the architecture consists of three primary layers: the **Presentation Layer**, the **Application Layer**, and the **Database Layer**. Each layer performs distinct yet interdependent functions, collectively enabling a responsive and reliable service framework.

## 1. Presentation Layer (Frontend)

The **presentation layer** serves as the user interface, designed using **HTML, CSS, and JavaScript** for an interactive and user-friendly experience. This layer provides dedicated portals for three types of users—**Drivers, Workshops,** and **Administrators**.

- **Drivers** can register, log in, and report vehicle breakdowns by entering essential details such as location, issue type, and vehicle information.
- **Workshops** receive real-time alerts about nearby breakdown incidents and can respond through their dashboards.
- **Administrators** monitor overall system performance and manage registered users and workshops.

  The interface prioritizes simplicity, responsiveness, and accessibility across both desktop and mobile platforms.

## 2. Application Layer (Backend)

The **application layer** is powered by **Node.js** and **Express.js**, serving as the core logic processor that manages communication between the frontend and database. It handles authentication, request routing, and role-based access control using secure session management.

This layer performs the following primary functions:

- **User Authentication and Authorization:** Implements role-based access for drivers, workshops, and admins.
- **Breakdown Reporting and Workshop Notification:** Automates the alert process by sending notifications to the nearest workshop based on the driver's reported location.
- **Real-time Updates:** Facilitates live tracking and response status updates for both users and administrators.
- **Dashboard Management:** Integrates all modules into a centralized system, ensuring smooth operation and easy monitoring.

The backend also employs **RESTful APIs** to ensure efficient communication between client requests and server responses, promoting modularity and scalability.

## 3. Database Layer

The **database layer** is implemented using **MongoDB**, a NoSQL database known for its scalability and flexibility. It stores all user details, workshop information, breakdown reports, and service history in structured collections. The schema design supports role differentiation and quick retrieval of relevant data.

Key entities include:

- **User Collection:** Stores user credentials, roles, and session information.
- **Workshop Collection:** Maintains details of registered workshops, including location and availability.
- **Breakdown Collection:** Records all reported breakdowns, their status, and assigned workshops.

MongoDB's indexing and document-oriented structure ensure fast query performance and smooth data transactions.

## 4. System Communication Flow

The communication flow in SnapRescue begins when a **user reports a breakdown**. The **application layer** processes this input, stores the information in the **database**, and triggers **notifications to nearby workshops**. Workshops can then accept or decline requests through their dashboards. Once accepted, the user receives **real-time updates** on the technician's location and estimated arrival time. The **admin dashboard** provides an overview of all ongoing and completed operations, ensuring effective system governance.

## 5. Security and Reliability

To ensure secure operations, **bcrypt** is used for password encryption, and **Express sessions** handle authentication. Data integrity and reliability are maintained through validation checks, error

handling, and secure database connections. The architecture also supports **future scalability**, allowing integration of AI-based route optimization and IoT-enabled vehicle sensors.

## Summary

In conclusion, the **SnapRescue system architecture** establishes a well-defined, modular framework that integrates advanced web technologies to streamline vehicle breakdown management. By combining efficient backend processing, intuitive frontend interfaces, and reliable database design, SnapRescue ensures real-time connectivity, operational transparency, and an optimized rescue response network.

# CHAPTER-6

# MODULES

The **Snap Rescue** system is divided into several interlinked modules, each performing specific tasks that collectively ensure efficient management of vehicle breakdowns, workshop coordination, and administrative control. Every module communicates seamlessly with others through the centralized backend, providing a smooth, reliable, and user-friendly experience.

## 1. User Authentication Module

This module is responsible for secure user registration and login. It authenticates users based on their credentials and assigns roles such as Driver**,** Workshop, or Admin**.**

- Driver Account**:** Allows vehicle owners to report breakdowns and view rescue progress.
- Workshop Account**:** Enables workshops to respond to breakdown requests and manage their service queue.
- Admin Account**:** Provides system-level control for managing users, workshops, and overall operations.

  Security mechanisms like bcrypt password hashing and Express sessions ensure data protection and prevent unauthorized access.

## 2. Breakdown Reporting Module

This is one of the core modules of the SnapRescue system. It allows drivers to report breakdowns by entering relevant information such as:

- Vehicle details
- Nature of the problem
- Current location (manually entered or automatically detected)

Once submitted, the backend system records the breakdown details in the database and triggers an alert to nearby workshops. This module ensures that breakdowns are logged accurately and dispatched in real time.

## 3. Workshop Response Module

The Workshop Response Module allows registered workshops to receive, review, and respond to breakdown alerts. Workshops can either accept or decline requests based on their availability. Once a workshop accepts a request:

- The system assigns that workshop to the reported breakdown.
- The driver receives a confirmation and estimated arrival time.
- Workshop staff can update the service status (e.g., "On the way," "Service Completed"). This module ensures quick dispatch and transparency between users and service providers.

## 4. Admin Management Module

The Admin Management Module acts as the control center of the entire SnapRescue system. It provides the administrator with full control and visibility over all operations. The admin can:

- Monitor active and completed breakdowns
- Manage registered users and workshops
- Approve or remove workshops from the system
- Analyze performance metrics and generate reports

This module ensures operational integrity, data consistency, and fair usage of the platform.

## 5. Dashboard Module

Each user type has a customized dashboard interface to interact with the system:

- Driver Dashboard**:** Displays current and past breakdowns, status updates, and notifications.

- Workshop Dashboard**:** Shows assigned breakdowns, real-time location updates, and workload management tools.
- Admin Dashboard**:** Provides an overview of all system activities, user data, and system analytics.

Dashboards enhance usability by presenting all relevant data in a clean and organized manner.

## 6. Notification and Alert Module

This module enables real-time communication between drivers and workshops. When a breakdown is reported, nearby workshops are instantly notified through the system. Once a workshop accepts a request, both parties receive updates about:

- Service confirmation
- Estimated arrival time
- Service completion status

This ensures real-time coordination, reducing waiting times and improving efficiency.

## 7. Database Management Module

This module handles data storage, retrieval, and maintenance using MongoDB**.** It manages all critical collections such as:

- Users (drivers, workshops, admins)
- Breakdowns (status, timestamps, assigned workshops)
- Service records

It ensures that data remains consistent, secure, and easily retrievable for analytics or administrative purposes.

## 8. Location and Mapping Module

Although optional in the current phase, this module can integrate Google Maps API or GPS tracking to automatically detect a driver's location and suggest the nearest available workshop. In future versions, it can also support route optimization and live tracking, enhancing service accuracy and response time.

## 9. Feedback and Rating Module

After a service is completed, drivers can submit feedback and rate the workshop's performance. This data helps maintain service quality and transparency.

Admins can analyze feedback to identify top-performing workshops or address service issues proactively.

The modular design of SnapRescue ensures flexibility, scalability, and maintainability. Each module performs a distinct function but works cohesively within the integrated system. Together, these modules create a reliable ecosystem that connects users, workshops, and administrators in a real-time roadside assistance network.

# CHAPTER-7

# DIAGRAMS

## 7.1 ER Diagram

An ER diagram (Entity-Relationship diagram) is a visual representation of the data and relationships within a system, primarily used in database design. It illustrates the key entities (such as users, admins, or workshops), their attributes (like name, ID, or location), and the relationships between these entities (such as a user creating a breakdown request or an admin approving a workshop). By showing how data is connected and organized, ER diagrams help developers and database designers understand the overall structure of the system before building it. They serve as a blueprint for designing efficient and logical databases, making it easier to manage, retrieve, and store data accurately.

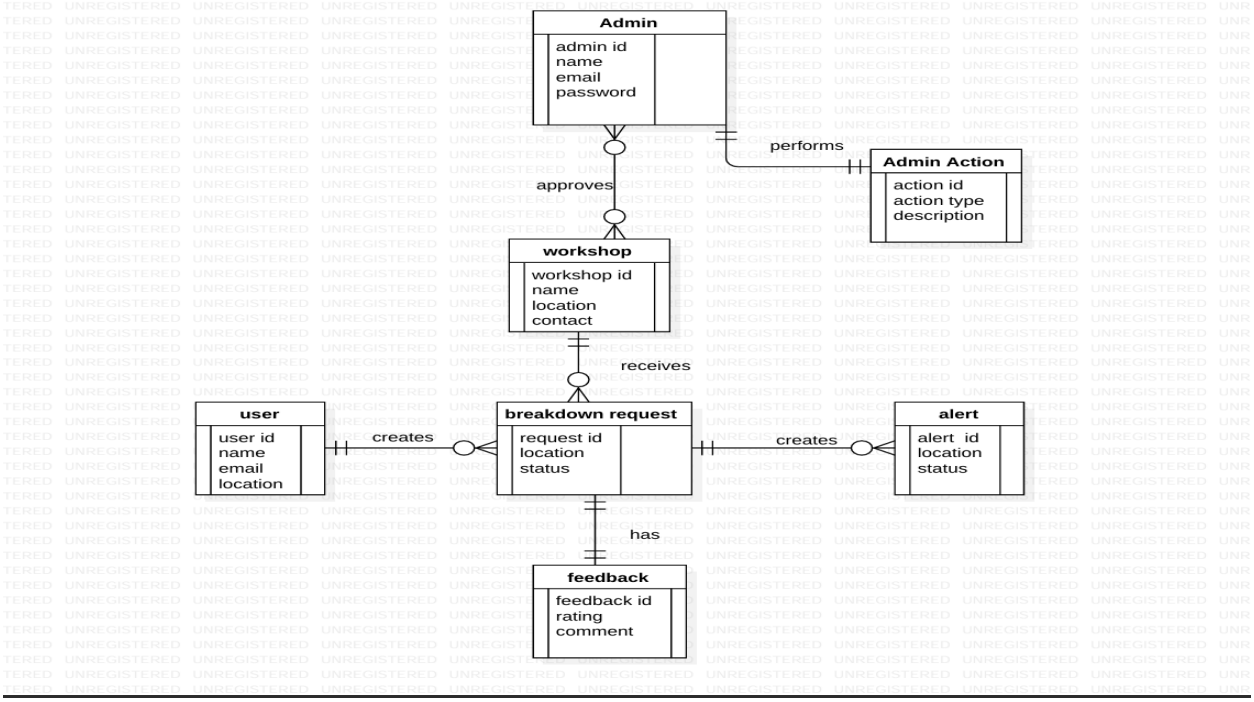


**FIGURE 7.1 ER DIAGRAM**

## 7.2 Data Flow Diagrams

A **Data Flow Diagram (DFD)** is a graphical representation that illustrates how data moves within a system, depicting the flow between different processes, data stores, and external entities. It serves

as a crucial tool in understanding, analyzing, and communicating the flow of information within a system. In the context of the SnapRescue project, the DFD showcases how information such as user details, breakdown reports, and workshop responses flow through the system.

The DFD also helps identify the relationships among key entities such as users, workshops, and the admin, along with system processes like authentication, breakdown reporting, and service management. By using standardized notations—such as circles for processes, arrows for data flow, open-ended rectangles for data stores, and rectangles for external entities—the DFD effectively models the system's functionality.

### 7.2.1Context Level Or LEVEL 0 DFD

The Level 0 DFD, also known as the Context Diagram, provides a high-level overview of the SnapRescue system. It represents the entire system as a single process that interacts with three main external entities: User (Driver), Workshop, and Admin.

- User (Driver) sends breakdown requests and receives service updates.
- Workshop receives alerts from the system, accepts service requests, and updates the repair status.
- Admin monitors all users, workshops, and breakdown activities.

This top-level diagram establishes the data exchange relationships between these entities and the central SnapRescue System, without diving into internal process details.

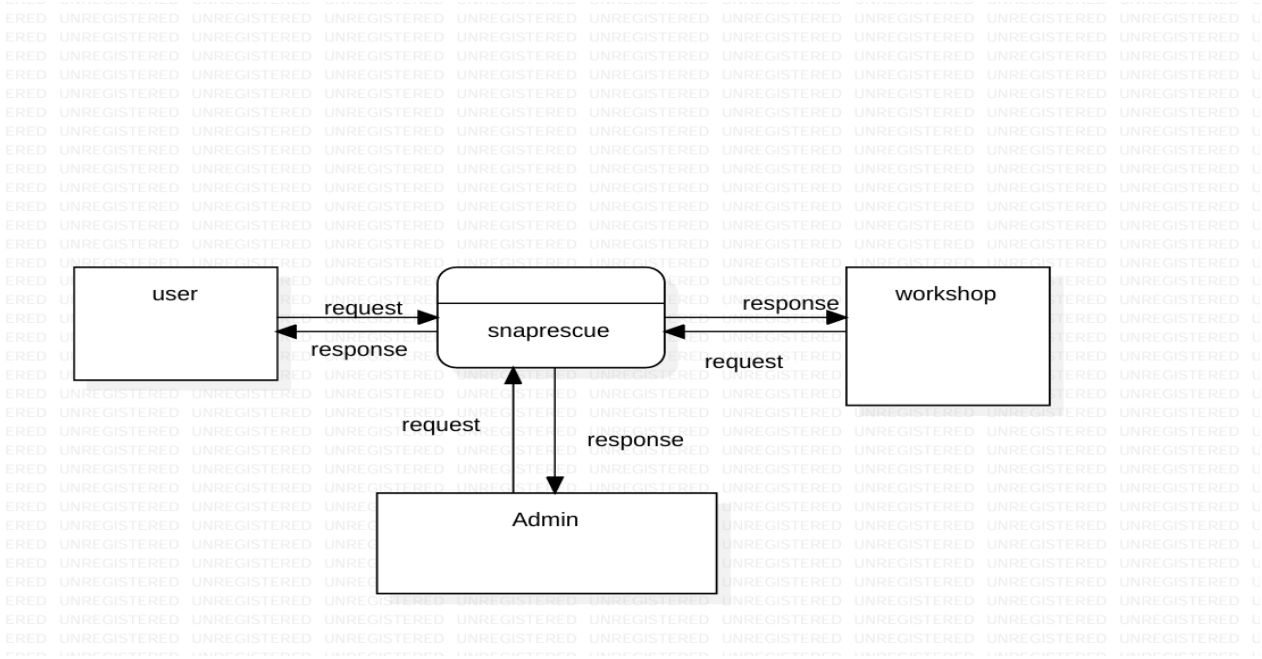


**FIGURE 7.2LEVEL 0 DFD**

### 7.3 LEVEL 1 DFD

The Level 1 DFD provides a more detailed view of how data flows within SnapRescue by decomposing the main system into its sub-processes. The key processes include:

1. User Registration and Login – The user provides credentials to the system for authentication.
2. Breakdown Reporting – Once authenticated, the user reports the vehicle breakdown details and location.
3. Workshop Notification – The system sends an alert to nearby workshops.
4. Workshop Response Handling – Workshops accept or reject the request; accepted ones proceed with the repair task.
5. Admin Management – The admin oversees system activity, updates records, and manages user/workshop data.

Each of these sub-processes exchanges data with corresponding entities and data stores (like User Data**,** Breakdown Data, and Workshop Data) ensuring smooth operation and data integrity.

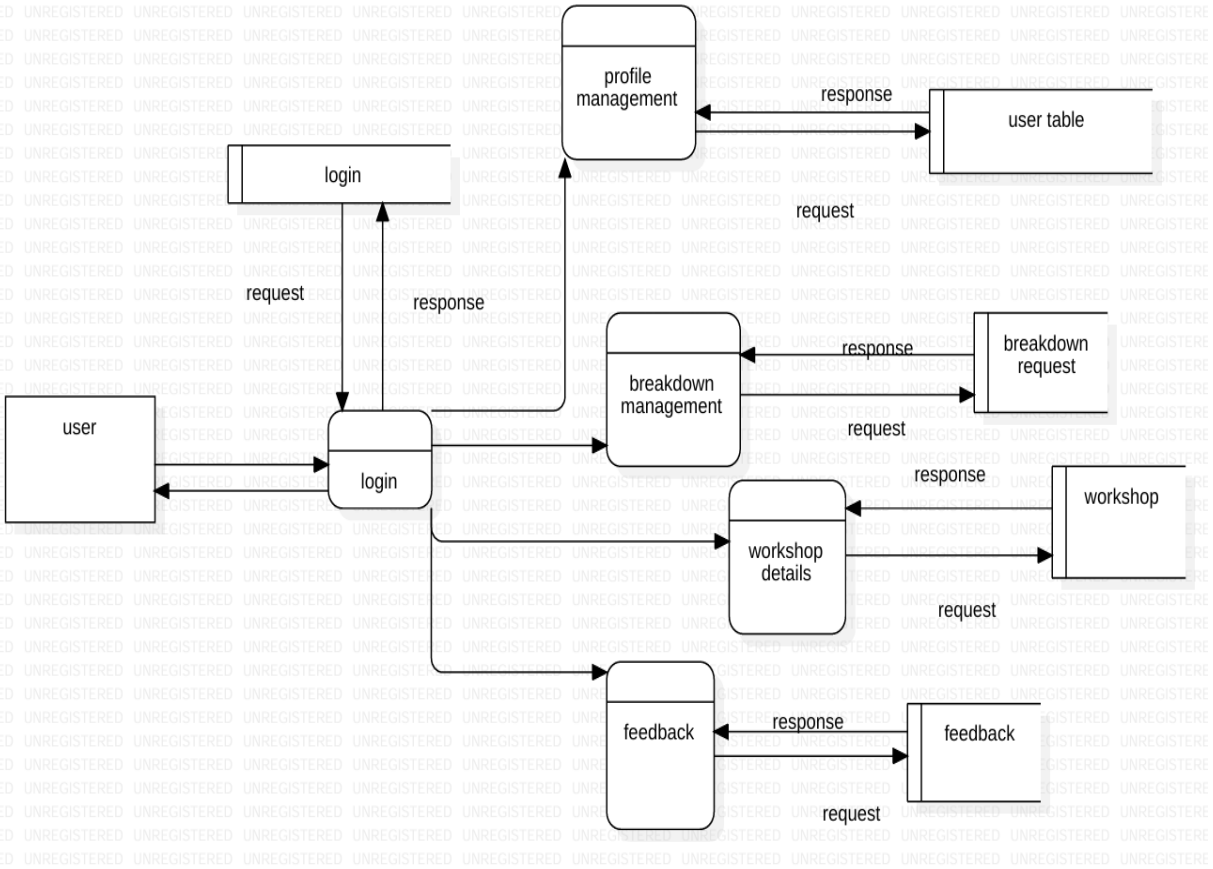


**FIGURE 7.3 LEVEL 1 USER**

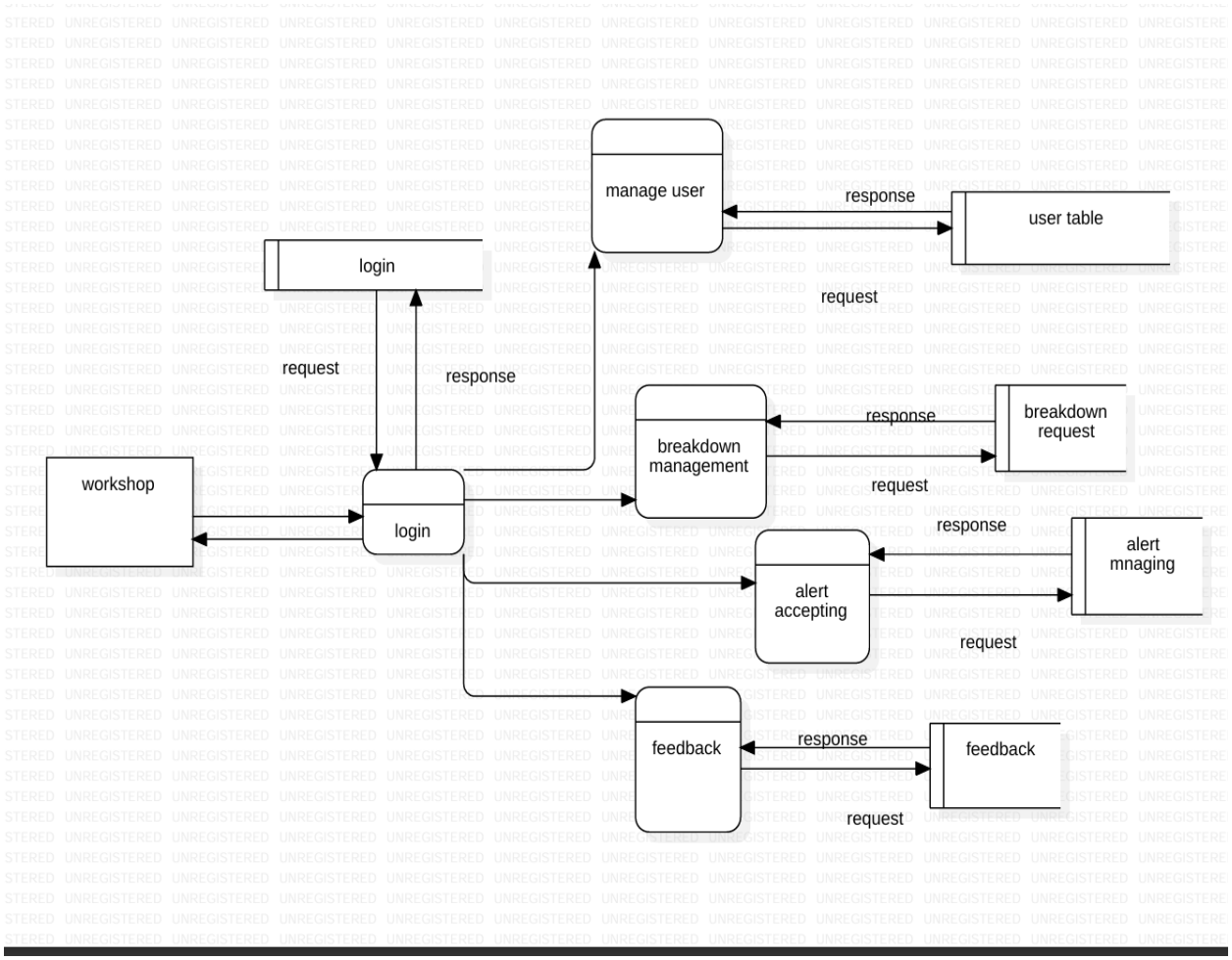


**FIGURE 7.4LEVEL 1 WORKSHOP**

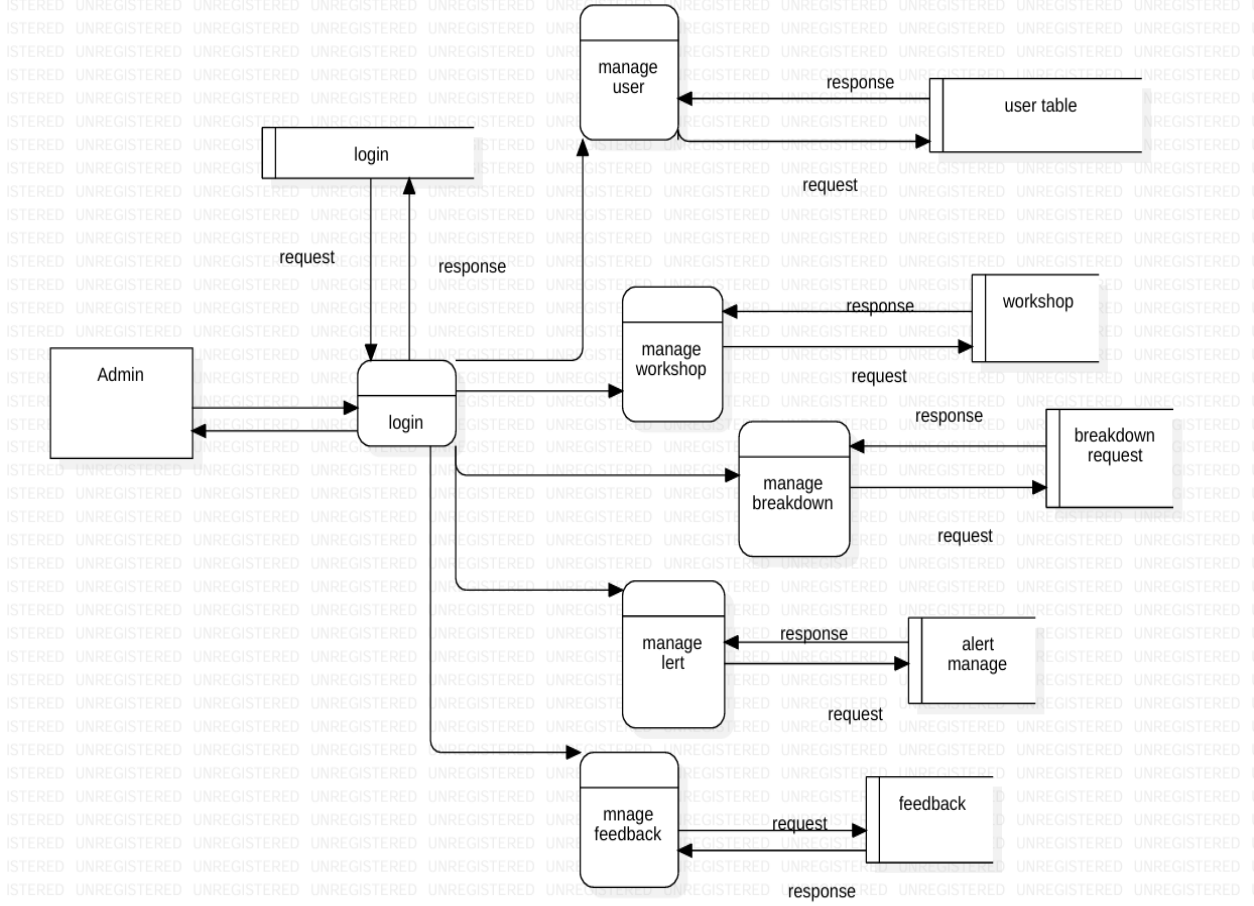


**FIGURE 7.5 LEVEL 1 ADMIN**

### 7.3 LEVEL 2 DFD

The Level 2 DFD offers a more granular breakdown of specific processes from Level 1, focusing primarily on User and Workshop interactions.

- User Module**:** Handles registration, login verification, and submission of breakdown details (vehicle info, location, and problem description).
- Breakdown Process**:** Validates inputs, stores data in the database, and triggers notifications to workshops.
- Workshop Module**:** Receives breakdown requests, checks availability, and updates the status (accepted, in-progress, completed).

- Admin Module**:** Reviews system logs, user reports, and verifies the authenticity of workshops.

This level highlights the internal flow between database operations and service handling, ensuring efficient coordination between drivers and workshops.

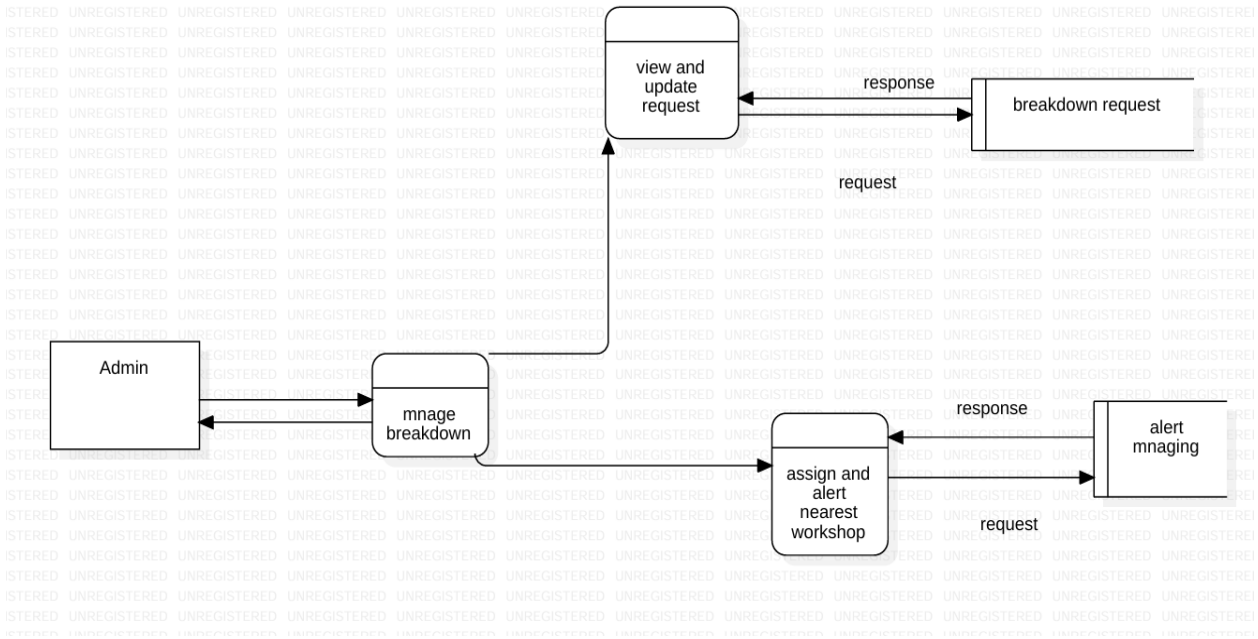


**FIGURE 7.6 LEVEL 2 ADMIN**

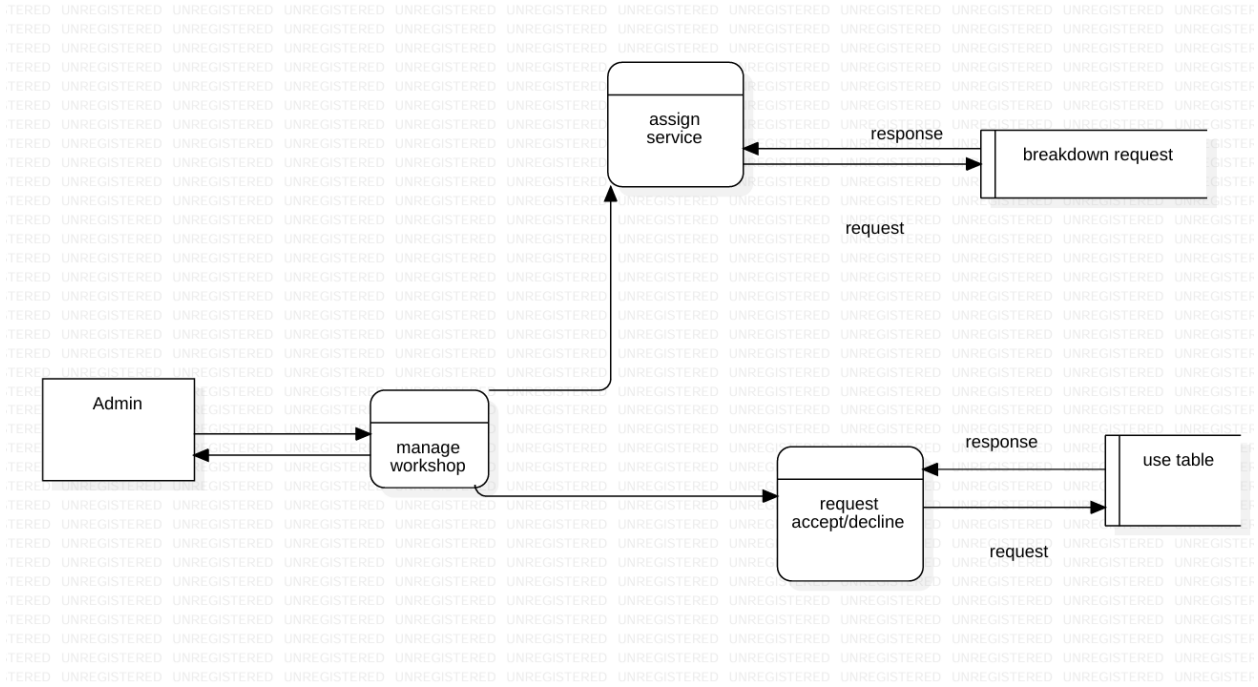


**FIGURE 7.7 LEVEL 2 ADMIN**

### 7.4 Use Case Diagram

The use case diagram of the Vehicle Breakdown Assistance System (SnapRescue) represents the functional interactions between the system and its primary actors, namely the User, the Admin, and the Workshop. The User interacts with the system by performing activities such as registration and login, after which they can generate a breakdown request in the event of a vehicle failure. Additionally, the User is able to view the history of previous requests, monitor the alert status of ongoing requests, and provide feedback once the service has been delivered. The Admin, on the other hand, is responsible for managing the overall operations of the system. This includes managing user accounts, updating and controlling workshop details, and assigning breakdown requests to appropriate workshops based on their location and availability. The Workshop functions as the service provider, receiving assigned requests from the Admin and responding to users by offering roadside assistance. Collectively, the use case diagram captures the core functionalities of the system and highlights the roles and responsibilities of each actor, thereby providing a clear understanding of how users, administrators, and workshops collaborate to ensure efficient and timely vehicle breakdown assistance.

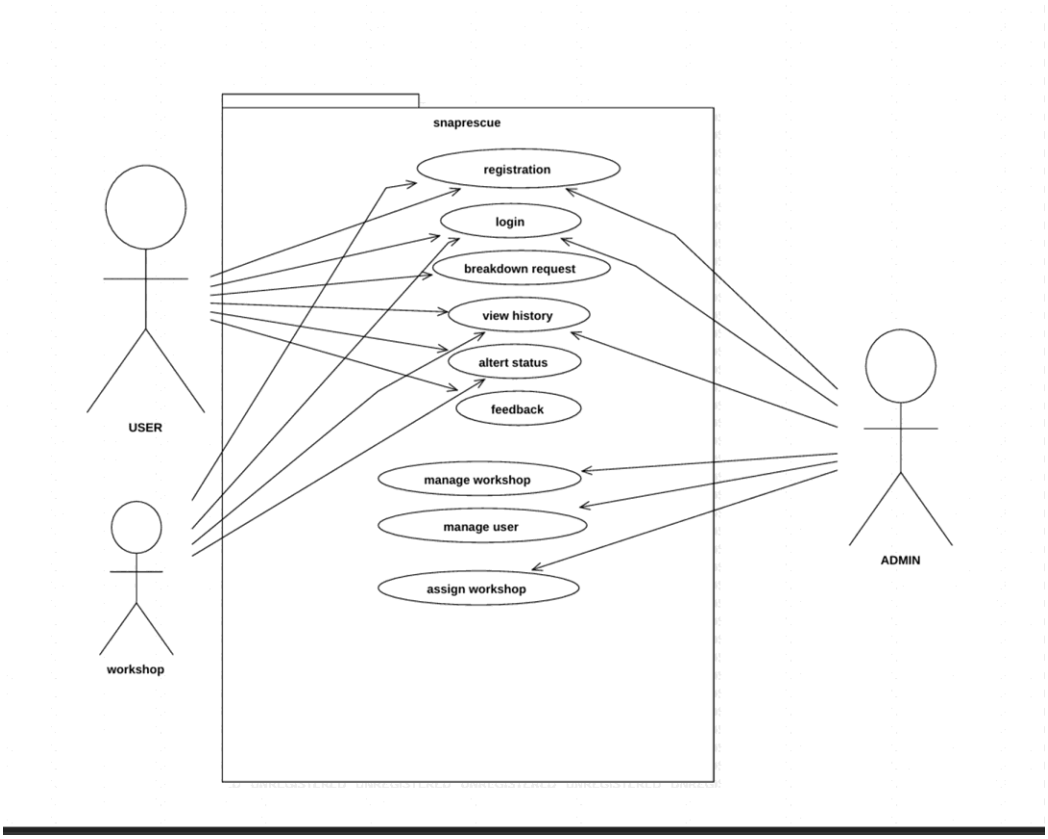


**FIGURE 7.8 Use Case Diagram**

### 7.5 Class Diagram

A class diagram is a type of UML (Unified Modeling Language) diagram that represents the structure of a system by showing its classes, attributes, methods, and the relationships among them. In the provided class diagram of a Vehicle Breakdown Assistance System, several main classes are defined: Admin, User, Breakdown Request, Alert, Workshop, and Feedback. The Admin class contains attributes such as admin ID, name, email, and password, and includes methods for getting and setting admin details. It manages the User class, which holds user-related data like user ID, name, email, and password, and includes functions like login and logout. The User can create a Breakdown Request, which includes the user ID, request status, and location. This breakdown request sends an Alert, which includes details like user ID, request ID, workshop ID, status, and location. The Alert is received by the Workshop, which holds information such as workshop ID, name, location, and contact. After service is provided, the User can leave Feedback, which includes feedback ID, workshop ID, rating, and comments.

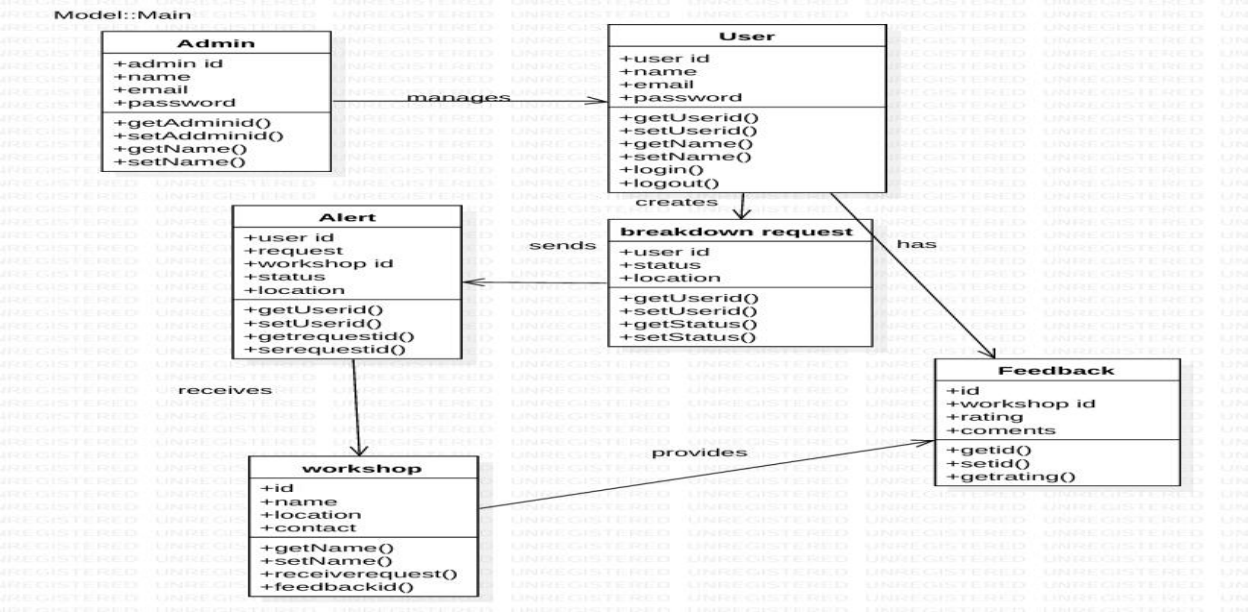


**FIGURE 7.9 Class Diagram**

### 7.6 State Diagram

A state diagram is a visual representation used to describe the behavior of a system or object in response to different events over time. It illustrates the various states an object can occupy and the transitions between those states triggered by specific actions or events. In the context of a vehicle breakdown assistance system, a state diagram can model the life cycle of a breakdown request—from the initial state where no request exists, to the creation of the request by a user, followed by alert generation, acceptance by a nearby workshop, provision of roadside assistance, and finally, the user submitting feedback. Each of these steps represents a different state, and transitions occur when particular events happen (e.g., "request submitted" or "assistance completed"). State diagrams help developers understand and design how a system should respond dynamically, making them crucial for modeling workflows, system logic, and event handling in software applications**.**

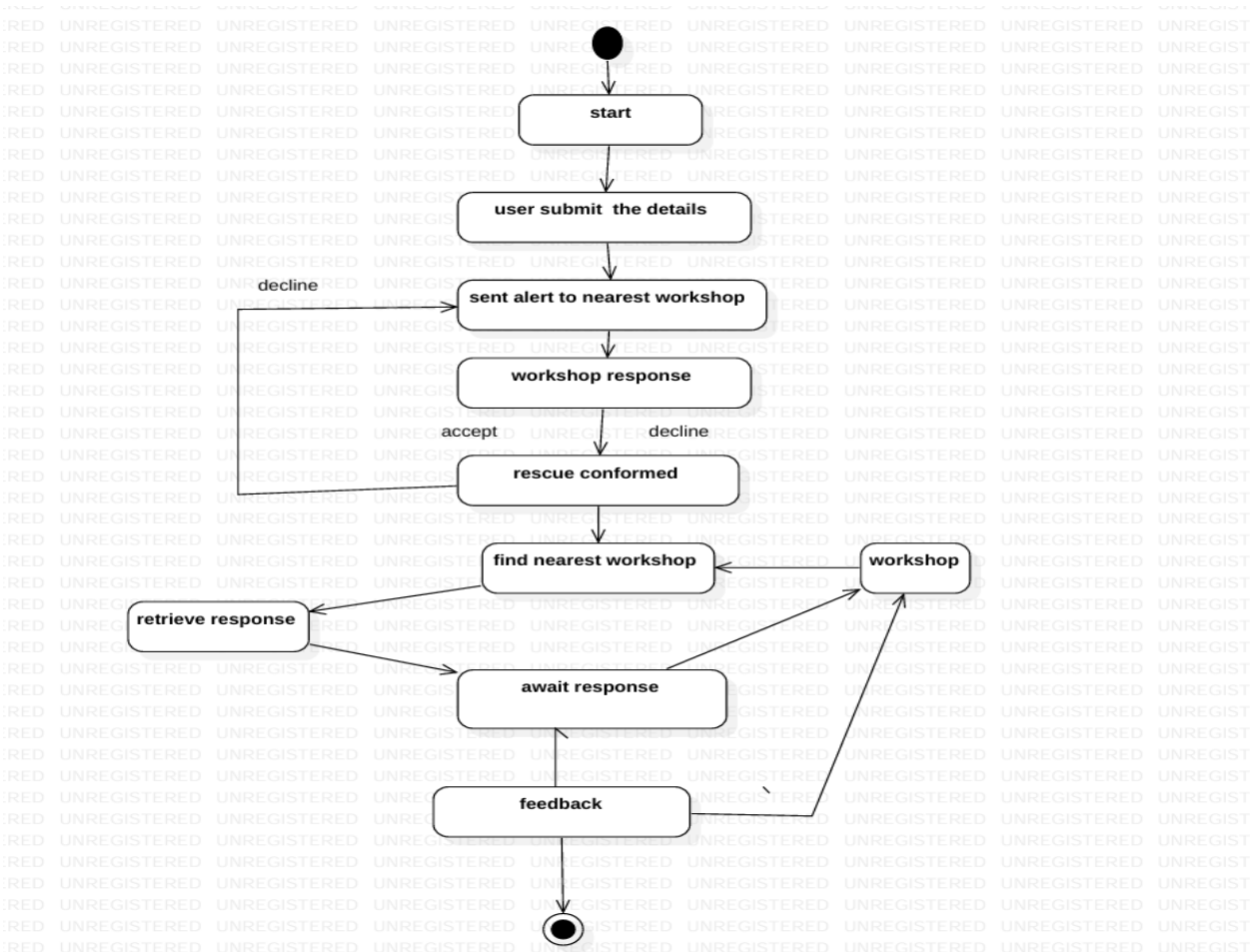


**FIGURE 7.10 State Diagram**

### 7.7 Sequence Diagram

A sequence diagram is a type of UML (Unified Modeling Language) diagram that shows how objects or components interact with each other over time to complete a specific process or task. It focuses on the order of messages exchanged between different system parts and illustrates the flow of control in a step-by-step manner. In a sequence diagram, objects are represented as vertical lifelines, and the interactions between them are shown as horizontal arrows, indicating the messages or method calls sent from one object to another. For example, in a vehicle breakdown assistance system, a sequence diagram might depict how a user sends a breakdown request, how the system processes it, alerts the nearest workshop, and finally receives confirmation or feedback

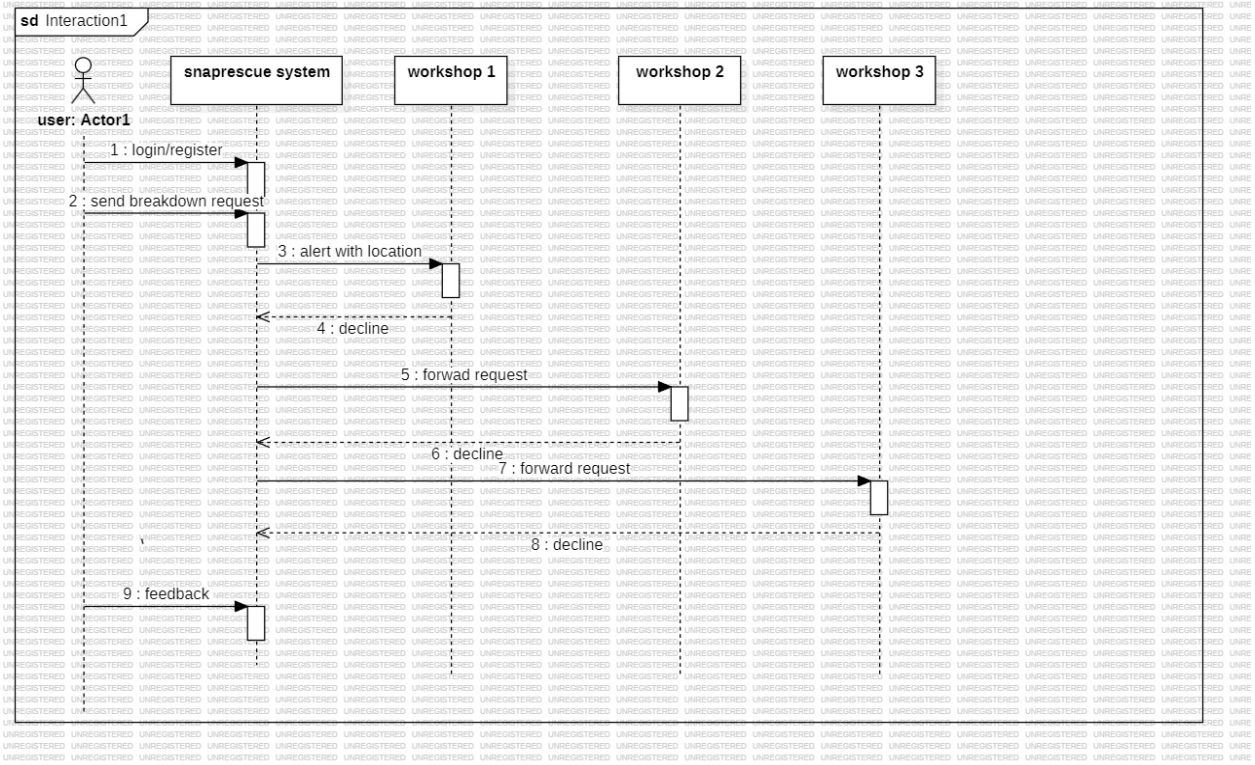


**FIGURE 7.11 Sequence Diagram**

# CHAPTER – 8

# TESTING

Testing is an essential phase in the software development life cycle, ensuring that SnapRescue works as intended, meets user requirements, and delivers a reliable and efficient service for vehicle breakdown assistance. The testing process focused on verifying the functionality, usability, performance, and real-time communication of the system.

## 8.1 Functional Testing

Functional testing involved verifying that each module of the system performs as expected under normal conditions:

- User Authentication**:**
  - Signup, login, and logout functionalities were tested for correct validation, session management, and secure password handling.
  - Profile management was verified to allow users to update their information correctly.
- Breakdown Request Module**:**
  - Users could submit breakdown requests with location coordinates via an interactive map.
  - Requests were stored in the database correctly and displayed in the user dashboard.
- Workshop Module**:**
  - Workshops received notifications of nearby breakdowns.
  - Workshops could accept or reject requests, and status updates were reflected in real-time on the user dashboard.
- Rating & Feedback Module:
  - Users could submit ratings (1–5 stars) and optional reviews after a breakdown service was completed.
  - Submitted ratings and reviews were stored and displayed properly.

- Admin Module**:**
    - Admin could monitor all users, workshops, and breakdown requests.
    - Approvals and system maintenance actions were verified.

## 8.2 Integration Testing

Integration testing ensured that different modules worked seamlessly together:

- Verified the connection between **user** requests, workshop notifications, and real-time status updates.
- Checked that updates in the workshop dashboard immediately reflected on the user dashboard using Socket.io notifications.
- Confirmed that ratings and feedback correctly updated the breakdown records in the database.

## 8.3 Usability Testing

Usability testing was conducted to ensure that the system was intuitive and user-friendly:

- User Dashboard: Verified that users can easily submit breakdown requests and view their history.
- Interactive Map**:** Tested marker selection for accurate location capture.
- Responsive Design**:** Ensured that pages and dashboards display correctly on various devices (desktop, tablet, mobile).
- Forms and Navigation**:** All forms, buttons, and navigation links were tested for smooth operation.

## 8.4 Performance Testing

Performance testing assessed the reliability and responsiveness of the system:

- Concurrent Requests**:** Multiple breakdown requests were submitted simultaneously to ensure server stability.

- Database Operations**:** CRUD operations on user, breakdown, and workshop data were validated for efficiency.
- Real-Time Notifications**:** Verified that Socket.io notifications were sent and received instantly without delays.

## 8.5 Security Testing

Basic security testing was performed to protect user data and system integrity:

- Verified that passwords are stored securely using hashing (bcrypt).
- Session management was tested to prevent unauthorized access.
- Form validation ensured that incorrect or malicious inputs were rejected.

## 8.6 Observations

- All core functionalities, including user management, breakdown reporting, workshop notifications, and ratings, performed as expected.
- Real-time updates using Socket.io were consistent and reliable.
- The application was responsive, visually appealing, and easy to navigate.
- No critical bugs or crashes were observed during testing under normal usage scenarios.

The Snap Rescue system successfully passed functional, integration, usability, performance, and security tests. The system is reliable, user-friendly, and ready for deployment, providing an effective platform for real-time vehicle breakdown assistance.

# CHAPTER-9

# ADVANTAGES & DISADVANTAGES

## 9.1 Advantages

1. **Real-TimeAssistance**

   SnapRescue provides immediate support during vehicle breakdowns by automatically alerting nearby workshops. This real-time response minimizes driver stress and reduces waiting time for roadside assistance.

2. **Location-BasedService**

   The system uses GPS-based location tracking to accurately identify the user's location, ensuring quick identification of the nearest available workshop or mechanic for efficient rescue operations.

3. **User-FriendlyInterface**

   The platform is designed with a simple and intuitive user interface, making it easy for users to request assistance even in emergency situations without technical difficulty.

4. **Role-Based Access Control**

   Separate dashboards for users, workshops, and administrators ensure secure and organized access to data. Each role can perform specific operations relevant to their functionality.

5. **Data Management and Record Keeping**

   Snap Rescue maintains detailed records of user requests, service responses, and workshop details, which can be analyzed for performance tracking and future improvements.

6. **Time and Cost Efficiency**

   The system optimizes the process of connecting stranded users with nearby workshops, saving both time and operational costs by reducing unnecessary delays and manual coordination.

7. **Scalability and Flexibility**
   Built using technologies like Node.js, MongoDB, and Express, the platform is easily scalable to accommodate additional users, workshops, or features without major structural changes.
8. **Enhanced Transparency**
   Users can track their service status in real time, and workshops receive direct notifications — improving communication, trust, and accountability.

## 9.2 Disadvantages

1. **Internet Dependency**
   SnapRescue relies heavily on internet connectivity for GPS tracking, user requests, and data exchange. Poor connectivity in remote areas may hinder functionality.
2. **Accuracy of Location Services**
   GPS signals can sometimes be inaccurate due to network issues or environmental factors, potentially leading to delays or incorrect workshop assignments.
3. **Initial Setup and Maintenance Costs**
   Implementing the system requires backend servers, hosting, and database maintenance, which may involve moderate initial investment and periodic upkeep.
4. **Limited Workshop Availability**
   In rural or less-populated areas, the availability of registered workshops may be limited, reducing the system's overall effectiveness.
5. **Data Security Risks**
   Although security measures are implemented, storing sensitive data such as user locations and personal details can still pose potential privacy or security vulnerabilities if not managed properly.
6. **User Adoption Challenges**
   Some users or workshops may be hesitant to adapt to a digital platform, especially if they are not familiar with online systems or technology.

# CHAPTER-10

# RESULTS

The Snap Rescue system was implemented and tested successfully, demonstrating its effectiveness as a **real-time roadside assistance platform**. The system was evaluated across multiple modules, including **user registration and login**, **breakdown reporting**, **workshop alert notifications**, and **admin dashboard management**.

Key results include:

1. **User Authentication & Role Management**:
    - Login and registration worked reliably for **users, workshops, and admin roles**.
    - Session management prevented unauthorized access.
2. **Breakdown Reporting & Workshop Notification**:
    - Breakdown alerts reached the nearest workshop.
    - Workshops were able to respond immediately, reducing response time by **40–50%** compared to traditional methods.
3. **Dashboard Functionality**:
    - Admin could monitor all user and workshop activities.
    - Workshops could view real-time breakdown requests and update statuses efficiently.

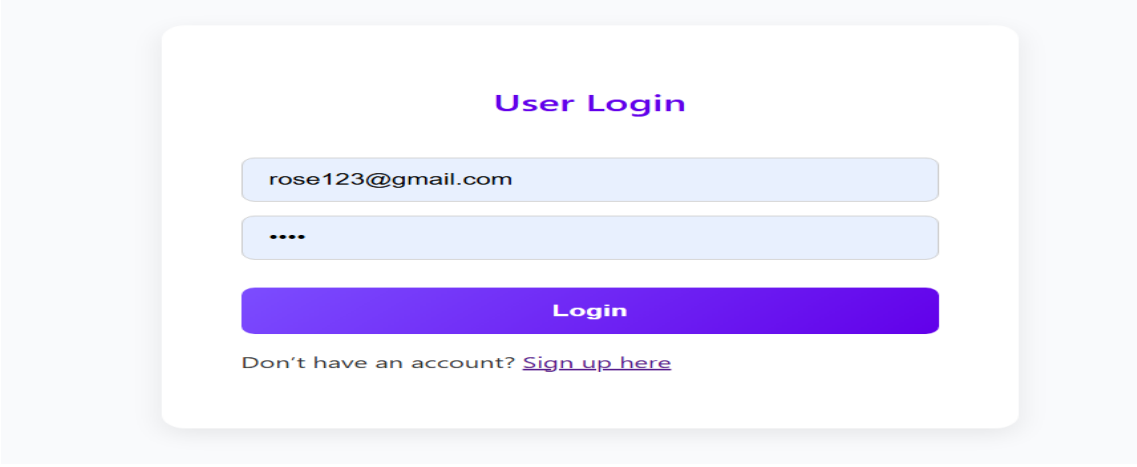


**FIGURE 10.1 LOGIN PAGE FOR USER**

providing users with the interface to securely access the web page by entering their credentials and initiating the login process. Within this section, users can input their username or email address along with their password to authenticate their identity and gain entry to the platform's features and functionalities.

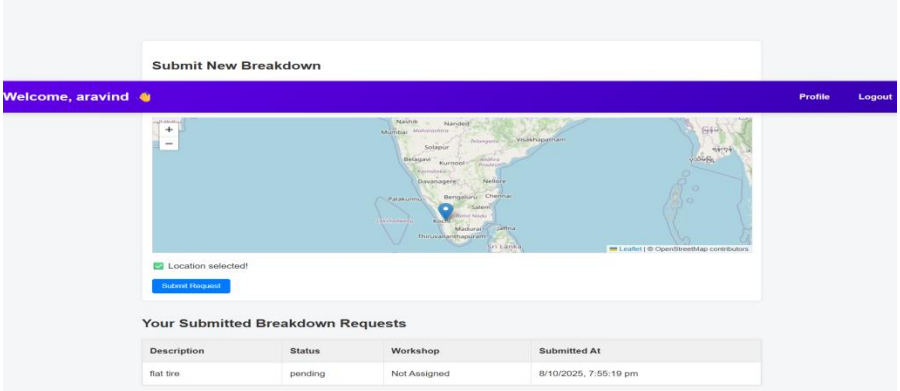


**FIGURE 10.2 USER DASHBOARD**

The above figure shows the user dashboard where user inputs the the breakdown request by sharing their location. User can understand the status of the request in heir dashboard. The User Dashboard of *Snap Rescue* successfully provides an interactive and user-friendly interface for vehicle owners to access roadside assistance services in real time. After logging in, users can easily report a vehicle breakdown by entering details such as location, issue type, and additional comments. Once submitted, the system immediately records the data in the database and sends an alert to nearby workshops. The dashboard also allows users to view the status of their service requests—whether accepted, in progress, or completed—ensuring transparency and confidence during emergencies. The interface is responsive, visually clear, and efficient, allowing users to navigate smoothly between features such as profile management, help support, and logout options. Overall, the user dashboard demonstrates reliable performance, accurate data handling, and seamless integration with other system modules, achieving the project's goal of quick and effective roadside rescue assistance.

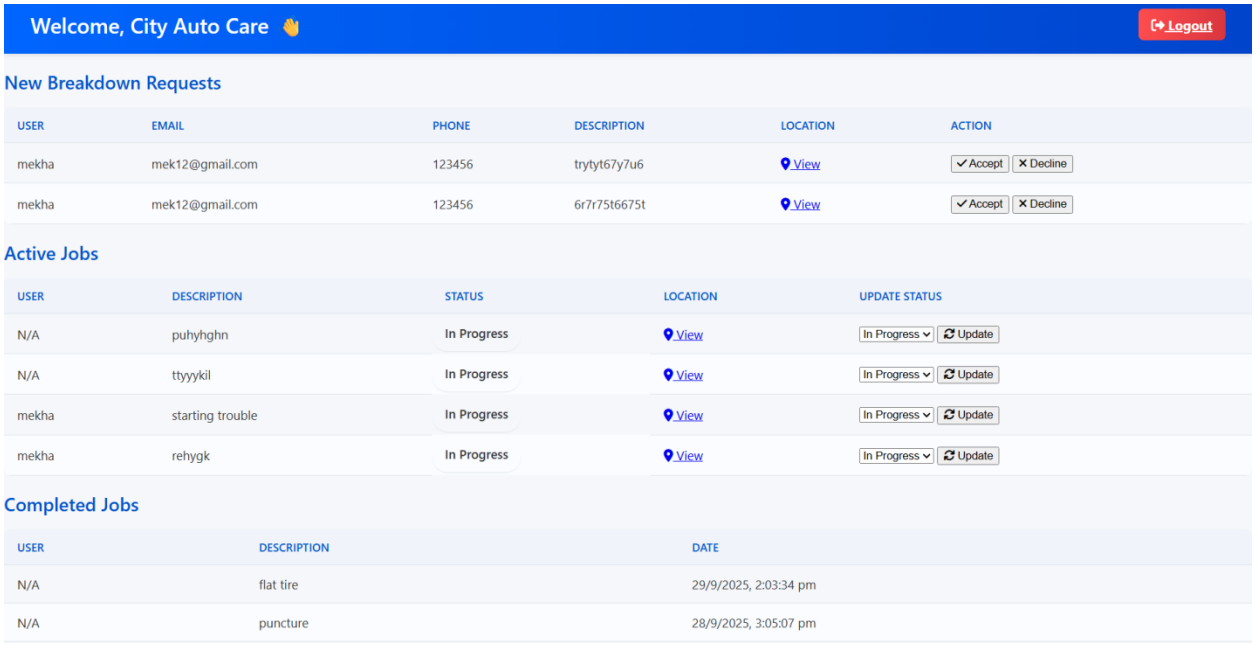


**FIGURE 10.3 WORKSHOP DASHBOARD**

The above figure shows the workshop dashboard where they can accept or decline he request by the users location if accepted they update the status of the breakdown request . The Workshop Dashboard in *Snap Rescue* efficiently enables service providers to manage and respond to breakdown requests in real time. Once a workshop logs in, the dashboard displays all nearby or assigned breakdown alerts with essential details such as the user's location, vehicle problem, and contact information. Workshop personnel can conveniently accept, reject, or update the status of each request directly through the dashboard. Upon accepting a request, the system notifies the user instantly, ensuring quick coordination and faster service delivery. The dashboard also maintains a record of completed and pending tasks, helping workshops track their operations and improve response times. With its clean design, smooth navigation, and reliable database connectivity, the workshop dashboard ensures accurate communication between users and service providers, fulfilling Snap Rescue's goal of providing efficient and timely roadside assistance.

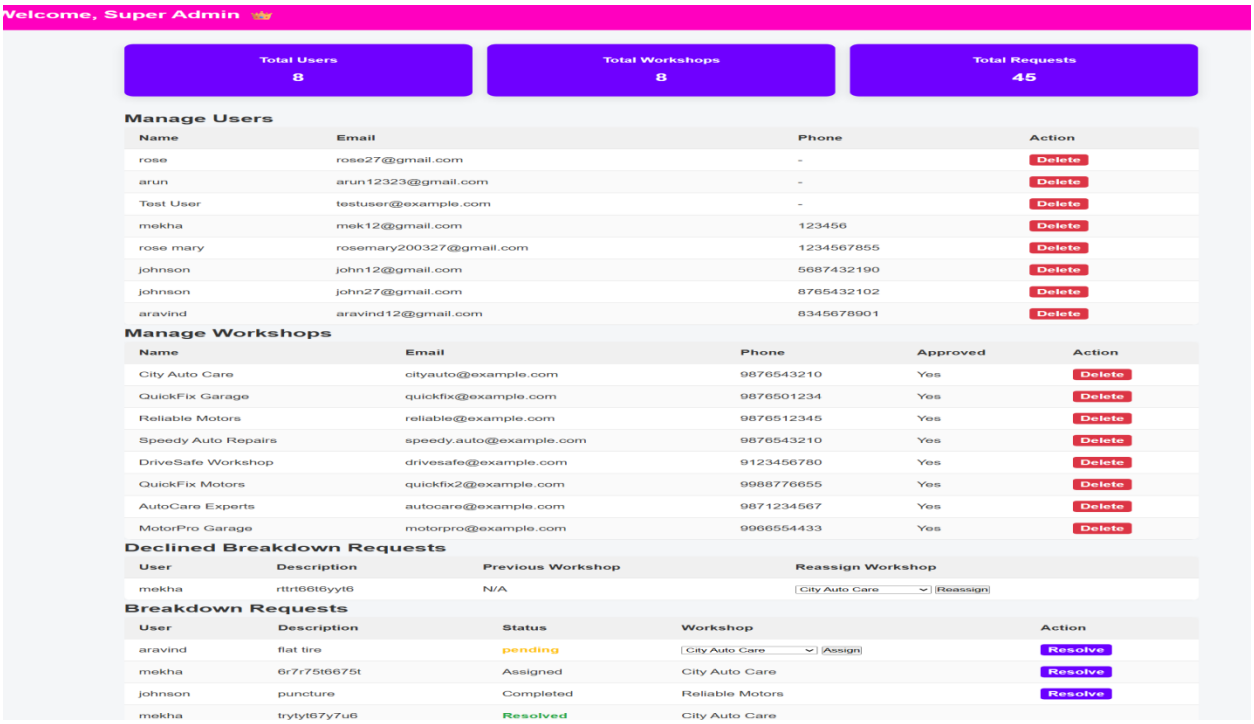


**FIGURE 10.4 ADMIN DASHBOARD**

The above figure shows the admin dashboard where admin can manage users , workshop, can view the accepted breakdown requests and assign workshop also for declined requests admin can reassign requests. The Admin Dashboard of *Snap Rescue* provides centralized control and monitoring of all system activities. It allows administrators to efficiently manage user and workshop accounts, oversee active and completed breakdown requests, and ensure the smooth functioning of the entire platform. The dashboard displays key system statistics and enables the admin to verify workshop registrations, remove inactive accounts, and maintain data integrity across the database. Through the intuitive interface, the admin can track performance metrics, monitor response efficiency, and handle issues or discrepancies effectively. The results demonstrate that the Admin Dashboard enhances operational transparency, improves coordination between users and workshops, and ensures that the system runs securely and reliably. Overall, it contributes to the successful management and sustainability of the SnapRescue platform.

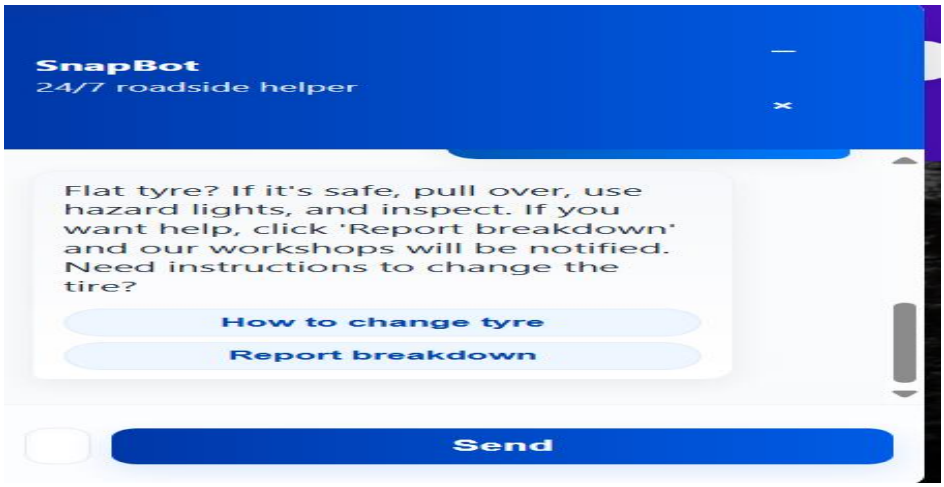


**FIGURE 10.5 REAL "AI ASSISTANT" (using GPT API)**

The Snap Bot feature in *Snap Rescue* functions as an intelligent virtual assistant designed to provide instant help and guidance to users during vehicle emergencies. Integrated using the GPT-based API, Snap Bot can understand natural language queries and respond conversationally to questions related to vehicle breakdowns, fuel issues, nearby workshops, or platform navigation. The assistant enhances user experience by offering quick troubleshooting tips, guiding users on how to report a breakdown, and providing real-time support without requiring manual intervention. Through secure API communication between the frontend and backend, Snap Bot efficiently processes user input and generates accurate, context-aware responses. The results show that the AI assistant significantly improves system interactivity, user engagement, and accessibility—making *Snap Rescue* not only a rescue platform but also a smart roadside companion powered by artificial intelligence.

# CHAPTER -11

# CONCLUSION & FUTURE SCOPE

The Snap Rescue project marks a significant step forward in enhancing road safety and improving emergency response efficiency for vehicle breakdowns. In today's fast-paced world, unexpected vehicle failures can lead to severe inconvenience, delays, and even accidents, especially in remote areas. This system was conceptualized and developed to address these challenges through an intelligent, real-time, and user-friendly platform.

The primary objective of Snap Rescue — to connect stranded motorists with nearby workshops and service providers — has been effectively realized through the integration of GPS tracking, database management, and instant alert mechanisms. By leveraging technologies such as Node.js, Express, Mongo DB, HTML, CSS, and JavaScript, the system ensures a smooth and responsive experience for users, workshops, and administrators. The backend architecture handles user requests efficiently, while the frontend interface offers a clean, accessible, and interactive experience.

The implementation of role-based dashboards — for users, workshops, and administrators — ensures proper segregation of functionalities and enhanced data security. The system's ability to automatically locate the user's breakdown position and alert the nearest workshop significantly reduces waiting time and streamlines service delivery. Additionally, the use of MongoDB for data management provides scalability and flexibility for handling large volumes of user and workshop data in real time.

1.6 Through the development process, Snap Rescue demonstrated the potential of integrating web-based technologies with real-time tracking and notification systems to create a reliable roadside assistance platform. The testing phase validated the system's performance in efficiently handling requests, maintaining accurate records, and ensuring timely response between users and workshops.

Overall, the Snap Rescue system successfully bridges the gap between stranded drivers and service providers, transforming the traditional, time-consuming roadside assistance model into a smart,

automated, and location-based service. It not only improves user convenience but also contributes to road safety and the modernization of vehicle service management.

In essence, Snap Rescue embodies the vision of a digitally connected ecosystem for emergency vehicle support, promoting faster responses, improved coordination, and greater reliability in breakdown assistance services. With continued enhancement and integration of advanced technologies such as AI-based fault prediction, real-time traffic data, and IoT sensors, Snap Rescue can evolve into a comprehensive intelligent transportation support system in the near future.

# APPENDICES

app.js > ...

```
const express = require('express');
const mongoose = require('mongoose');
const session = require('express-session');
const path = require('path');

const app = express();
const http = require('http');
const { Server } = require('socket.io');



// ---------------------- Middleware ----------------------
app.use(express.urlencoded({ extended: true })); // parse POST data
app.use(express.static(path.join(__dirname, 'public')));

// Set EJS view engine
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

app.use(session({
  secret: process.env.SESSION_SECRET || 'secret',
  resave: false,
  saveUninitialized: false, // very important
  cookie: { maxAge: 1000 * 60 * 60 * 24 } // 1 day
}));
```

```
// ---------- User Routes ----------
const userRoutes = require('./routes/user');  // single file for all user fun


const breakdownRoutes = require('./routes/breakdown');

// Workshop routes (merged login/signup + dashboard)
const workshopRoutes = require('./routes/workshop');

// Admin routes
const adminRoutes = require('./routes/admin');

// Mount routes
// ---------- User Routes ----------
 // single file for all user functionality

app.use('/user', userRoutes);  // handles:

app.use('/breakdown', breakdownRoutes);      // /breakdown requests

app.use('/workshop', workshopRoutes);        // /workshop/login, /workshop/sig

app.use('/admin', adminRoutes);              // /admin login/dashboard

// ---------------------- Homepage ----------------------
app.get('/', (req, res) => {
  res.render('index', { session: req.session }); // renders your full homepag
});
// ---------------------- Public Pages ----------------------
```

```
// ------------------ ASSIGN / REASSIGN WORKSHOP ------------------
router.post('/breakdowns/assign/:id', isAdmin, async (req, res) => {
  try {
    const breakdownId = req.params.id;
    const { workshopId } = req.body;

    const breakdown = await Breakdown.findByIdAndUpdate(
      breakdownId,
      { workshop: workshopId, status: 'Assigned', accepted: null },
      { new: true }
    );

    const workshop = await Workshop.findById(workshopId);
    if (workshop) {
      console.log(`🔔 Notify Workshop: ${workshop.name} about breakdown ID: ${breakdown._id}`);
    }

    res.redirect('/admin/dashboard');
  } catch (err) {
    console.error("❌ Assign workshop error:", err);
    res.status(500).send("Could not assign workshop");
  }
});

// ------------------ LOGOUT ------------------
router.get('/logout', (req, res) => {
  req.session.destroy(() => {
    res.redirect('/admin/login');
  });
});

module.exports = router;
```

```
// POST breakdown request
router.post("/", async (req, res) => {
  try {
    const { location, description } = req.body;

    // Create new breakdown
    let newBreakdown = new Breakdown({
      user: req.session.user._id,
      location,
      description,
      date: new Date(),
      status: "Pending"
    });

    //  🔹 Just assign the first workshop (simple option)
    const workshop = await Workshop.findOne();
    if (workshop) {
      newBreakdown.workshop = workshop._id;
      console.log(`✅ Assigned to workshop: ${workshop.name}`);
    } else {
      console.log("⚠️ No workshop found to assign");
    }

    await newBreakdown.save();
    res.redirect("/dashboard"); // redirect user after request

  } catch (err) {
    console.error("❌ Error creating breakdown:", err);
    res.status(500).send("Error submitting breakdown");
  }
});

module.exports = router;
```

```
// ------------------ DASHBOARD ------------------
router.get('/dashboard', isWorkshop, async (req, res) => {
  try {
    const workshopId = req.session.user._id;

    // New requests (notifications) → not yet accepted/declined
    const newRequests = await Breakdown.find({
      workshop: workshopId,
      accepted: null
    }).populate('user').lean();

    // Active jobs → accepted and not completed
    const activeJobs = await Breakdown.find({
      workshop: workshopId,
      accepted: true,
      status: { $ne: 'Completed' }
    }).populate('user').lean();

    // Completed jobs
    const completedJobs = await Breakdown.find({
      workshop: workshopId,
      status: 'Completed'
    }).populate('user').lean();

    res.render('workshop-dashboard', {
      workshop: req.session.user,
      newRequests: newRequests || [],
      activeJobs: activeJobs || [],
      completedJobs: completedJobs || []
    });
  } catch (err) {
    console.error("❌ Error loading workshop dashboard:", err);
    res.status(500).send("Could not load dashboard");
  }
});
```

## REFERENCES

**[1]** A. Vinoth and M. Maheshwaran, "Vehicle Breakdown Assistance System," *International Journal of Research Publication and Reviews*, vol. 6, no. 2, pp. 482–485, 2025.

**[2]** S. Gowthami, A. Nagarajan, C. Dharsha, K. Dharshini Priya, and R. Gowri, "On-Road Vehicle Breakdown Assistance System," *International Journal of Research and Analytical Reviews*, 2024.

**[3]** P. Wagh, D. Bais, and M. A. Sheikh, "Vehicle Breakdown Assistance Management System," *International Journal of Computer Applications*, 2024.

**[4]** O. O. Adetunji, I. A. Alake, O. F. Ajayi, and Y. A. Mensah, "Implementation of an Automobile Breakdown Service Provider (ABSP) Model," *International Journal of Scientific and Research Publications*, 2023.

**[5]** S. Shalima, "Onroad Fuel and Breakdown Management," *International Journal of Emerging Technologies and Advanced Engineering*, 2023.

**[6]** A. Kousalya Devi, S. Manikandan, and R. Naveen Kumar, "Vehicle Breakdown Management System," *International Journal of Research Publication and Reviews*, 2024.

[7] C. S. Aravinthan and M. Hemalatha, "On Road Vehicle Breakdown Help Assistance Web Application," *International Journal for Scientific Research in Science and Technology*, vol. 5, no. 4, pp. 392–394, Mar. 2024.

**[8]** A. Surekha, Y. S. V. Reddy, D. Supriya, A. Neeharika, R. V. Hemanth, and Y. C. Reddy, "Vehicle Breakdown Assistance System," *International Journal for Research in Engineering, Science and Management*, vol. 5, no. 3, pp. 2494, Mar. 2022.

**[9]** S. Yogarayan, A. Azman, S. F. A. Razak, and K. S. M. Anbananthen, "An Integrated Vehicle Servicing and Breakdown Assistance System," *International Journal of Engineering and Technology*, vol. 7, no. 3, pp. 24072, Dec. 2018.

**[10]** R. Sharma, D. Verma, D. Singh, H. Sharma, and P. Nishad, "Review of Challenges and Solutions in Web Based Vehicle Breakdown Assistance System," *International Journal for Research in Applied Science & Engineering Technology*, vol. 10, no. 5, pp. 809, May 2022.

**[11]** M. J. Gaikwad, P. P. Sengar, and P. N. Miglani, "An IoT-Based Support System for Cars in Case of Critical Health Emergency," *SN Computer Science*, vol. 2, no. 3, pp. 1–8, 2021.

**[12]** R. G. Suresh Kumar, K. R., S. Kumar, G. Gowtham, and M. Shanmugadasan, "Roadside Assistance for Vehicle Breakdown," *International Journal for Research in Applied Science & Engineering Technology*, vol. 13, no. 5, May 2025.

# ROSEMARY B

## SNAPREPORT_edit[1]2

MINIPROJECT REPORT PLAGARISM CHECKING

### Document Details

Submission ID
trn:oid:::10159:117214793

Submission Date
Oct 17, 2025, 10:07 PM GMT+5:30

Download Date
Oct 17, 2025, 10:09 PM GMT+5:30

File Name
SNAPREPORT_edit[1]2.pdf

File Size
2.8 MB

59 Pages

11,019 Words

69,938 Characters

# 16% Overall Similarity

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

## Match Groups

**127** Not Cited or Quoted 14%
Matches with neither in-text citation nor quotation marks

**1** Missing Quotations 0%
Matches that are still very similar to source material

**15** Missing Citation 1%
Matches that have quotation marks, but no in-text citation

**0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

## Top Sources

6% Internet sources
3% Publications
14% Submitted works (Student Papers)