

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI – 590018



Project Report

on

DATA BASE MANAGEMENT SYSTEMS GOVERNMENT SCHEME TOOLKIT

Submitted in partial fulfillment for the award of degree of

**Bachelor of Engineering in
Artificial Intelligence and Machine Learning**

Submitted by –

**USN
1RN21AI004
1RN21AI005**

**Name
AASHUTOSH SHARAN
AASTHA KRISHNA**



RNS INSTITUTE OF TECHNOLOGY

(AICTE Approved, VTU Affiliated and NAAC 'A+' Accredited)

(UG programs – CSE, ECE, ISE, EIE and EEE are Accredited by NBA up to 30.6.2025)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098

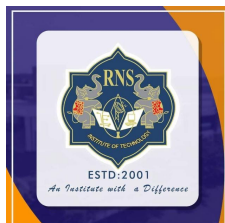
Department of CSE(AI&ML)

2023 – 24

RNS INSTITUTE OF TECHNOLOGY

(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)
(UG programs – CSE, ECE, ISE, EIE and EEE are Accredited by NBA up to 30.6.2025)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098

Department of CSE(AI&ML)



CERTIFICATE

Certified that the Project entitled **Government Scheme Toolkit** carried out by **Mr. Aashutosh Sharan (1RN21AI004)** and **Ms. Aastha Krishna (1RN21AI005)** a student of 5th Semester BE, **RNS Institute of Technology** in partial fulfillment for the Bachelor of Engineering in **Artificial Intelligence and Machine Learning** of **Visvesvaraya Technological University**, Belagavi during the year **2023-24**. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report. The Mini-Project report has been approved as it satisfies the academic requirements in respect of Database Management System with Mini Project Laboratory prescribed for the said Degree.

Course Teacher

Ms. Pooja M
Assistant Professor
Department of CSE(AI&ML)
RNSIT, Bengaluru

HoD

Dr. Harsha S
Department of (CSE-AI&ML)
RNSIT, Bengaluru

Name & Signature

Examiner 1:

Examiner 2:

ACKNOWLEDGEMENT

At the very onset, we would like to place on record our gratitude to all those people who have helped us in making this project work a reality. Our institution has played a paramount role in guiding us in the right direction.

We would like to profoundly thank **Sri. Satish R Shetty**, Chairman, RNS Group of Institutions, Bengaluru, for providing such a healthy environment for the successful completion of this project work.

We would like to thank **Dr. M K Venkatesha**, Director, RNSIT, Bengaluru for providing the necessary facilities to carry out this work.

We would like to express our gratitude to **Dr. Ramesh Babu H S** Principal, RNSIT, Bengaluru for his constant encouragement.

We are extremely grateful to **Dr. Harsha S**, Head of the Department of Artificial Intelligence and Machine Learning, for having accepted to guide me in the right direction with all his wisdom.

We are expressing our sincere thanks to our Course Teacher **Pooja M** Assistant Professor, Department of Artificial Intelligence and Machine Learning for her constant encouragement that motivated us for the successful completion of this project work.

Last but not the least, we are thankful to all the teaching and non-teaching staff members of the Artificial Intelligence and Machine Learning department for their encouragement and support throughout this work.

Signature

Aashutosh Sharan – 1RN21AI004

Aastha Krishna – 1RN21AI005

TABLE OF CONTENTS

Chapter No.	CONTENTS	PAGE NO.
	Acknowledgement	I
	Table of Contents	II
	Figure Names	III
1	Introduction	1-4
	1.1 Overview of project	
	1.2 Motivation	
	1.3 Problem Definition	
	1.4 Objectives	
2	Process	5-7
	2.1 Methodology	
	2.2 Block diagram	
	2.3 ER Diagram	
3	SQL Queries	8-9
	3.1 Creation	
4	Code	10-17
	4.1 Code	
5	Outputs	18-19
	5.1 Outputs of MySQL	
6	Output of Webpages	20-22
	6.1 Web Interfaces	
7	Conclusion and Future Enhancements	23-25
	References	26

FIGURE NAMES

Figure No.	Figure Name	PAGE NO.
2.1	Block Diagram	7
2.2	ER Diagram	7
5.1	User Table Details	18
5.2	All details in table scheme_details	18
5.3	All details in table scheme_name	19
5.4	Tables Present	19
5.5	Flights Present at the time	12
5.6	Number of Ground Staff present	12
6.1	Home Page	20
6.2	User Registration Page	20
6.3	Login Page	21
6.4	Forgot Password Page	21
6.5	User Page	22
6.6	Admin Page	22

Chapter-1

INTRODUCTION

1.1 Overview of Project

In an era of rapid digitalization and governmental initiatives aimed at enhancing citizen welfare, the advent of a government scheme toolkit stands as a beacon of accessibility and efficiency. This innovative platform serves as a comprehensive repository, consolidating a myriad of governmental schemes into a single, user-friendly interface. Through this toolkit, users gain unparalleled access to a wealth of information, enabling them to navigate and leverage diverse government programs effortlessly.

The government scheme toolkit emerges as a powerful tool in the arsenal of both citizens and policymakers alike. For citizens, it represents a gateway to explore the array of welfare schemes, subsidies, and benefits extended by the government. By providing a centralized hub for scheme discovery, users are empowered to make informed decisions regarding their participation in various programs tailored to their specific needs and circumstances. Simultaneously, for policymakers, this toolkit serves as a mechanism for promoting transparency and accountability in governance, facilitating better tracking of scheme utilization and effectiveness.

The versatility of the government scheme toolkit extends beyond mere scheme exploration. With robust search functionalities and categorization features, users can swiftly filter through an extensive database of schemes based on parameters such as eligibility criteria, sector, and geographical region. This not only streamlines the process of scheme discovery but also ensures that individuals and organizations can easily identify opportunities for assistance or collaboration, fostering inclusivity and socioeconomic development across diverse segments of society.

1.2 Motivation

The motivation behind the development of the government scheme toolkit stems from a profound desire to address longstanding challenges related to the accessibility and awareness of governmental welfare programs. In many countries, despite the presence of numerous schemes aimed at improving the lives of citizens, there often exists a significant gap between the availability of these resources and the awareness among the intended beneficiaries. This gap can be attributed to various factors, including the decentralized nature of scheme information, lack of coordination between government departments, and limited outreach efforts.

1.3 Problem Definition

The problem that the government scheme toolkit seeks to address is the lack of centralized and accessible information regarding various government welfare programs and schemes. In many countries, government initiatives aimed at providing assistance, subsidies, and benefits to citizens are dispersed across multiple departments and agencies, leading to fragmentation and complexity in scheme discovery and utilization. As a result, individuals and organizations often struggle to navigate the labyrinth of schemes, leading to inefficiencies, missed opportunities, and inequitable access to vital resources.

Furthermore, there is a pervasive issue of limited awareness among the intended beneficiaries regarding the existence and eligibility criteria of government schemes. This lack of awareness exacerbates disparities in access to services and benefits, particularly among marginalized and vulnerable populations who may stand to benefit the most from government assistance. Without a centralized platform for scheme information, individuals may remain unaware of opportunities for financial support, education assistance, healthcare services, or entrepreneurial incentives available to them.

Moreover, the absence of a comprehensive repository for government schemes hampers policymakers' ability to assess the effectiveness and reach of these programs. Without robust data on scheme utilization and impact, policymakers face challenges in optimizing resource allocation, refining program design, and

addressing gaps in service delivery. This lack of transparency and accountability undermines the efficacy of governance and impedes efforts to achieve inclusive and sustainable development goals.

In essence, the problem statement for the government scheme toolkit revolves around the need to centralize, streamline, and democratize access to information about government welfare programs. By addressing this fundamental challenge, the toolkit aims to enhance transparency, promote citizen empowerment, and facilitate more efficient and equitable distribution of government resources, ultimately contributing to improved social welfare and economic prosperity for all members of society.

1.4 Objectives

- **Centralization of Scheme Information:** The primary objective of the toolkit is to create a centralized repository that aggregates information about various government welfare programs and schemes. By consolidating scheme details from different departments and agencies into a single platform, the toolkit aims to simplify the process of scheme discovery and access for citizens, businesses, and other stakeholders.
- **Enhancing Accessibility and Awareness:** Another key objective is to enhance the accessibility and awareness of government schemes among the target audience. Through user-friendly interfaces, intuitive search functionalities, and informative content, the toolkit seeks to ensure that individuals and organizations can easily identify relevant schemes based on their needs, eligibility criteria, and geographical location.
- **Promoting Transparency and Accountability:** The toolkit aims to promote transparency and accountability in governance by providing comprehensive information about scheme objectives, eligibility criteria, application procedures, and outcomes. By making scheme data readily accessible to policymakers, researchers, and the general public, the toolkit facilitates

greater scrutiny, evaluation, and improvement of government programs.

- **Facilitating Citizen Engagement and Participation:** A key objective of the toolkit is to facilitate greater citizen engagement and participation in government schemes. By empowering individuals with the knowledge and resources to avail themselves of available benefits and services, the toolkit aims to promote active citizenship, foster community development, and enhance social cohesion.
- **Supporting Policy Design and Decision-Making:** The toolkit serves as a valuable tool for policymakers and administrators by providing insights into scheme utilization patterns, demographic trends, and effectiveness metrics. By leveraging data analytics and reporting features, the toolkit supports evidence-based policy design, resource allocation, and decision-making processes, thereby enhancing the overall efficiency and impact of government interventions.

Chapter-2

PROCESS

2.1 Methodology

The methodology for implementing the government scheme toolkit involves several key steps and strategies to ensure its effectiveness and success. The methodology can be outlined as follows:

1. Needs Assessment: The first step involves conducting a thorough needs assessment to understand the specific requirements, challenges, and priorities related to accessing government schemes. This may include stakeholder consultations, surveys, and analysis of existing data to identify gaps in scheme information, user preferences, and technological infrastructure.

2. Database Development: Based on the needs assessment, a comprehensive database of government schemes is developed, encompassing various sectors such as healthcare, education, agriculture, social welfare, and entrepreneurship. This database includes detailed information about each scheme, including its objectives, eligibility criteria, application process, benefits, and contact information.

3. Technology Selection and Development: The toolkit is developed using appropriate technology platforms and frameworks to ensure scalability, security, and user-friendliness. This may involve the use of web development technologies, databases, application programming interfaces (APIs), and data visualization tools to create an interactive and dynamic user interface.

4. Content Curation and Management: High-quality content is curated and regularly updated to ensure the accuracy, relevance, and completeness of scheme information. Content management systems (CMS) are employed to facilitate easy content creation, editing, and publishing by authorized administrators, ensuring timely dissemination of scheme updates and

announcements.

5. User Experience Design: The toolkit is designed with a user-centric approach, focusing on intuitive navigation, clear layout, and responsive design to enhance the overall user experience. Usability testing and feedback mechanisms are incorporated to gather user insights and iteratively improve the toolkit's usability and functionality.

6. Outreach and Training: Extensive outreach and training programs are conducted to promote awareness and usage of the toolkit among target audiences, including citizens, government officials, non-profit organizations, and businesses. Training sessions, workshops, and educational materials are provided to familiarize users with the toolkit's features, functionalities, and benefits.

7. Monitoring and Evaluation: Robust monitoring and evaluation mechanisms are implemented to track the usage, impact, and effectiveness of the toolkit over time. Key performance indicators (KPIs) are defined to measure metrics such as user engagement, scheme uptake, user satisfaction, and overall socio-economic impact, enabling continuous improvement and refinement of the toolkit.

8. Collaboration and Partnerships: Collaborations with government agencies, non-governmental organizations (NGOs), academic institutions, and private sector stakeholders are established to foster synergies, leverage resources, and maximize the reach and impact of the toolkit. Partnerships may involve data sharing agreements, joint promotional campaigns, and co-development initiatives to enhance the toolkit's functionality and relevance.

By following this comprehensive methodology, the government scheme toolkit can be effectively developed, deployed, and maintained to fulfill its objectives of enhancing scheme accessibility, promoting transparency, and empowering citizens to avail themselves of government benefits and services.

2.2 Block Diagram



Fig 2.1 Block Diagram

2.3 ER Diagram:

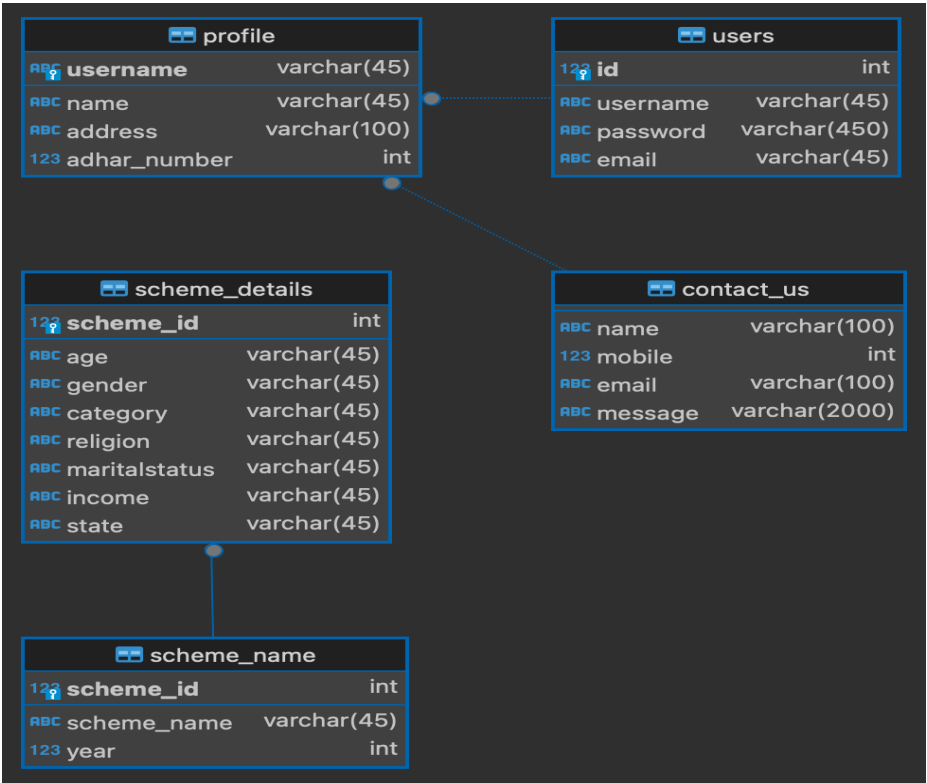


Fig 2.2 ER Diagram

Chapter-3

SQL QUERIES

3.1 Creation:

1. **contact_us:**

```
CREATE TABLE `contact_us` (  
  `name` varchar(100) DEFAULT NULL,  
  `mobile` int DEFAULT NULL,  
  `email` varchar(100) DEFAULT NULL,  
  `message` varchar(2000) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

2. **profile:**

```
CREATE TABLE `profile` (  
  `username` varchar(45) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `address` varchar(100) DEFAULT NULL,  
  `adhar_number` int NOT NULL,  
  PRIMARY KEY (`username`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

3. **scheme_details:**

```
CREATE TABLE `scheme_details` (  
  `scheme_id` int NOT NULL,  
  `age` varchar(45) DEFAULT NULL,  
  `gender` varchar(45) DEFAULT NULL,  
  `category` varchar(45) DEFAULT NULL,  
  `religion` varchar(45) DEFAULT NULL,  
  `maritalstatus` varchar(45) DEFAULT NULL,  
  `income` varchar(45) DEFAULT NULL,  
  `state` varchar(45) DEFAULT NULL,
```

```
PRIMARY KEY (`scheme_id`),  
CONSTRAINT `schemeid` FOREIGN KEY (`scheme_id`)  
REFERENCES `scheme_name` (`scheme_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

4. **scheme_name:**

```
CREATE TABLE `scheme_name` (  
  `scheme_id` int NOT NULL,  
  `scheme_name` varchar(45) NOT NULL,  
  `year` int DEFAULT NULL,  
  PRIMARY KEY (`scheme_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

5. **users:**

```
CREATE TABLE `users` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `username` varchar(45) DEFAULT NULL,  
  `password` varchar(450) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_0900_ai_ci DEFAULT NULL,  
  `email` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT  
CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

4.1 Code

```
const express = require('express');
const path = require('path');
const mysql = require('mysql');
const bcrypt = require('bcrypt');
const bodyParser = require('body-parser');
const nodemailer = require('nodemailer');

const app = express();
const port = 3000;

// Middleware
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

// Serve static files from the 'public' directory
app.use(express.static(path.join(__dirname, 'public')));

// Add another route to serve files from the 'public/main' directory
app.use('/main', express.static(path.join(__dirname, 'public', 'main')));

// MySQL Connection
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'aashu123', // Replace 'your_password' with your MySQL password
  database: 'dbms'
});

db.connect((err) => {
  if (err) {
    console.error('Error connecting to MySQL database:', err);
    return;
  }
  console.log('Connected to MySQL database');
});

// Serve static files from the 'public' directory
app.use(express.static(path.join(__dirname, 'public')));
```

```
// Define route for the root URL
app.get('/', (req, res) => {
  console.log(`[${new Date().toLocaleString()}] GET request received for /`);
  res.sendFile(path.join(__dirname, 'public', 'index.html'));
});

// Login route
// Login route
app.post('/login', (req, res) => {
  console.log(`[${new Date().toLocaleString()}] POST request received for /login`);
  const { usernameOrEmail, password } = req.body;
  console.log(`[${new Date().toLocaleString()}] Username or Email: ${usernameOrEmail}`);
  db.query('SELECT * FROM users WHERE username = ? OR email = ?',
[usernameOrEmail, usernameOrEmail], (error, results) => {
    if (error) {
      console.error(`[${new Date().toLocaleString()}] Error querying database:`,
error);
      res.status(500).json({ error: 'Internal server error' });
    } else {
      if (results.length > 0) {
        const user = results[0];
        bcrypt.compare(password, user.password, (err, isMatch) => {
          if (err) {
            console.error(`[${new Date().toLocaleString()}] Error comparing
passwords:`, err);
            res.status(500).json({ error: 'Internal server error' });
          } else {
            if (isMatch) {
              console.log(`[${new Date().toLocaleString()}] Successful login for user:
${user.username}`);
              if (user.email === 'resetschemedbms@gmail.com') {
                res.redirect('/main/admin_dashboard.html'); // Redirect to admin
dashboard
              } else {
                res.redirect('/main/dashboard.html'); // Redirect to regular user
dashboard
              }
            } else {
              console.log(`[${new Date().toLocaleString()}] Invalid password for
user: ${user.username}`);
              res.status(401).json({ error: 'Invalid username or password' });
            }
          }
        });
      } else {
        console.log(`[${new Date().toLocaleString()}] User not found for username
or email: ${usernameOrEmail}`);
        res.status(401).json({ error: 'Invalid username or password' });
      }
    }
  });
});
```



```
    }
  }
});
});

// Signup route
app.post('/signup', (req, res) => {
  console.log(`[${new Date().toLocaleString()}] POST request received for /signup`);
  const { username, email } = req.body;
  console.log(`[${new Date().toLocaleString()}] Username: ${username}, Email: ${email}`);
  const saltRounds = 10;
  bcrypt.hash(req.body.password, saltRounds, (err, hashedPassword) => {
    if (err) {
      console.error(`[${new Date().toLocaleString()}] Error hashing password:`, err);
      res.status(500).json({ error: 'Internal server error' });
    } else {
      db.query('INSERT INTO users (username, email, password) VALUES (?, ?, ?)', [username, email, hashedPassword], (error, results) => {
        if (error) {
          console.error(`[${new Date().toLocaleString()}] Error inserting user into database:`, error);
          res.status(500).json({ error: 'Internal server error' });
        } else {
          console.log(`[${new Date().toLocaleString()}] User signed up successfully: ${username}`);
          res.redirect('main/dashboard.js');
        }
      });
    }
  });
});

const showMessage = (res, message) => {
  res.status(200).json({ message: message });
};

app.post('/forgot', async (req, res) => {
  console.log(`[${new Date().toLocaleString()}] POST request received for /forgot`);
  const { email } = req.body;
  console.log(`[${new Date().toLocaleString()}] Email found: ${email}`);

  try {
    // Generate temporary password
    const temporaryPassword = generateTemporaryPassword();
```

```
// Hash the temporary password before storing it in the database
bcrypt.hash(temporaryPassword, 10, async (err, hashedPassword) => {
  if (err) {
    console.error(`[${new Date().toLocaleString()}] Error hashing
temporary password:`, err);
    return res.status(500).json({ error: 'Internal server error' });
  }

  try {
    // Update user's password in the database with the temporary password
    await updateUserPassword(email, hashedPassword);

    // Send email with the temporary password
    sendResetPasswordEmail(email, temporaryPassword);
    console.log(`[${new Date().toLocaleString()}] Temporary password
updated in the database for email: ${email}`);

    showMessage(res, 'Password reset email sent');
  } catch (error) {
    console.error(`[${new Date().toLocaleString()}] Error updating user
password or sending email:`, error);
    res.status(500).json({ error: 'Error resetting password' });
  }
});

} catch (error) {
  console.error(`[${new Date().toLocaleString()}] Error generating temporary
password:`, error);
  res.status(500).json({ error: 'Error resetting password' });
}
});

// Function to generate a temporary password
function generateTemporaryPassword() {
  // Generate a random temporary password (you can customize the length and
  characters as needed)
  return Math.random().toString(36).slice(2);
}

// Function to update user's password in the database
function updateUserPassword(email, hashedPassword) {
  return new Promise((resolve, reject) => {
    db.query('UPDATE users SET password = ? WHERE email = ?',
[hashedPassword, email], (error, results) => {
      if (error) {
        console.error(`[${new Date().toLocaleString()}] Error updating user
password:`, error);
        reject(error);
      } else {
        console.log(`[${new Date().toLocaleString()}] Temporary password updated
```

```
in the database for email: ${email}`);
    resolve();
  }
});
});
}

// Function to send reset password email
function sendResetPasswordEmail(email, temporaryPassword) {
  // Create Nodemailer transporter with SMTP and app password
  const transporter = nodemailer.createTransport({
    service: 'gmail',
    auth: {
      user: 'resetschemedbms@gmail.com', // Your Gmail email address
      pass: 'yhwd airl evjh rrcp' // Your app password
    }
  });

  // Send email
  transporter.sendMail({
    from: 'resetschemedbms@gmail.com', // Your Gmail email address
    to: email,
    subject: 'Password Reset',
    text: `Your temporary password for resetting the password is:
    ${temporaryPassword}`
  }, (error, info) => {
    if (error) {
      console.error(`[${new Date().toLocaleString()}] Error sending email:`, error);
    } else {
      console.log(`[${new Date().toLocaleString()}] Password reset email sent:`,
        info.response);
    }
  });
}

app.post('/search', (req, res) => {
  console.log(`[${new Date().toLocaleString()}] POST request received for
  /search`);
  console.log(`[${new Date().toLocaleString()}] Request body:`, req.body);

  const { age, gender, category, religion, maritalstatus, income, state } = req.body;

  console.log(`[${new Date().toLocaleString()}] Parameters extracted:`, { age,
    gender, category, religion, maritalstatus, income, state });

  // Construct the SQL query based on user inputs
  const sql = `
  SELECT s.scheme_name, sd.*
  FROM scheme_name s
```

```
JOIN scheme_details sd ON s.scheme_id = sd.scheme_id
WHERE
  sd.state = ? AND
  sd.religion = ? AND
  sd.maritalstatus = ? AND
  sd.income = ? AND
  sd.gender = ? AND
  sd.category = ? AND
  sd.age <= ?

`;
```

```
console.log(`[${new Date().toLocaleString()}] SQL Query: ${sql}`);
console.log(`[${new Date().toLocaleString()}] Parameters: ${[state, religion,
maritalstatus, income, gender, category, age].join(', ')}');

db.query(sql, [state, religion, maritalstatus, income, gender, category, age],
(error, results) => {
  if (error) {
    console.error(`[${new Date().toLocaleString()}] Error querying database for
schemes:`, error);
    res.status(500).json({ error: 'Internal server error' });
  } else {
    if (results.length > 0) {
      console.log(`[${new Date().toLocaleString()}] Found ${results.length}
schemes matching the criteria`);
      res.status(200).json({ schemes: results });
    } else {
      console.log(`[${new Date().toLocaleString()}] No schemes found matching
the criteria`);
      res.status(404).json({ error: 'No schemes found matching the criteria' });
    }
  }
});
});
```

```
// Handle POST request to add a new scheme to the database
app.post('/add-scheme', (req, res) => {
  console.log(`[${new Date().toLocaleString()}] POST request received for /add-
scheme`);
```

```
  // Extract data from the request body
  // Extract data from the request body
  const { id, name, year, age, gender, category, religion, maritalstatus, income, state
} = req.body;
```

```
  // Construct SQL query to insert the new scheme name into the scheme_name
```

```
table
const insertSchemeNameSQL = `
  INSERT INTO scheme_name (scheme_id, scheme_name)
  VALUES (?, ?)
`;

// Execute the SQL query to insert the new scheme name into the scheme_name
table
db.query(insertSchemeNameSQL, [id, name, year], (error, results) => {
  if (error) {
    console.error(`[${new Date().toLocaleString()}] Error inserting scheme
name into database:`, error);
    res.status(500).json({ error: 'Internal server error' });
  } else {
    console.log(`[${new Date().toLocaleString()}] Scheme name added to the
database`);

    // Once the scheme name is added successfully, insert scheme details into
scheme_details table
    const insertSchemeDetailsSQL = `
      INSERT INTO scheme_details (scheme_id, age, gender, category,
religion, maritalstatus, income, state)
      VALUES (?, ?, ?, ?, ?, ?, ?, ?)
    `;

    // Execute the SQL query to insert scheme details into scheme_details table
    db.query(insertSchemeDetailsSQL, [id, age, gender, category, religion,
maritalstatus, income, state], (error, results) => {
      if (error) {
        console.error(`[${new Date().toLocaleString()}] Error inserting
scheme details into database:`, error);
        res.status(500).json({ error: 'Internal server error' });
      } else {
        console.log(`[${new Date().toLocaleString()}] Scheme details added
to the database`);
        res.status(200).json({ message: 'Scheme added successfully' });
      }
    });
  }
});

// Handle contact form submission
app.post('/contact', (req, res) => {
  console.log(`[${new Date().toLocaleString()}] POST request received for
/contact`);

  const { name, mobile, email, message } = req.body;

  console.log(`[${new Date().toLocaleString()}] Contact form submitted with
```

```
data:');
console.log('Name:', name);
console.log('Mobile:', mobile);
console.log('Email:', email);
console.log('Message:', message);

// Insert form data into the contact_us table
const insertContactQuery = 'INSERT INTO contact_us (name, mobile, email,
message) VALUES (?, ?, ?, ?)';
db.query(insertContactQuery, [name, mobile, email, message], (error, results)
=> {
  if (error) {
    console.error(`[${new Date().toLocaleString()}] Error inserting contact form
data into database:`, error);
    res.status(500).json({ error: 'Internal server error' });
  } else {
    console.log(`[${new Date().toLocaleString()}] Contact form data inserted into
database`);
    // Send success message back to client
    res.status(200).json({ message: 'Contact form submitted successfully' });
  }
});
});

app.listen(port, () => {
  console.log(`[${new Date().toLocaleString()}] Server is running on
http://localhost:${port}`);
});
```

OUTPUTS

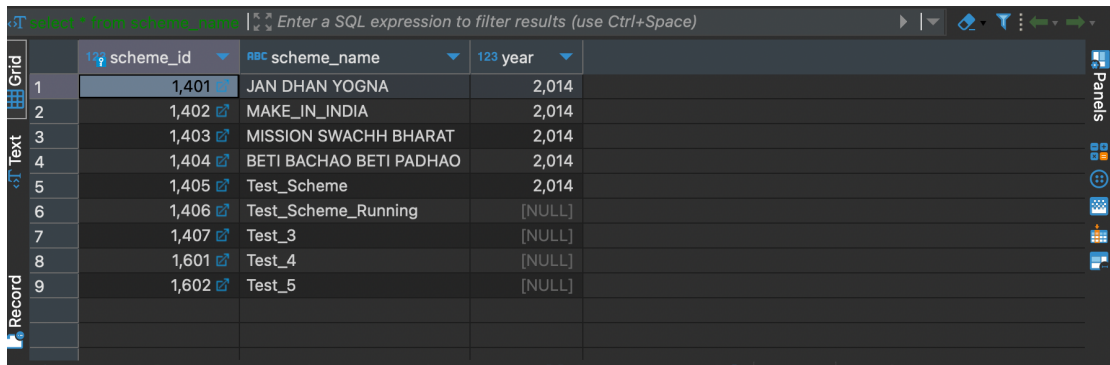
5.1 Output (MySQL)

[illegible]

Fig 5.1 user table details

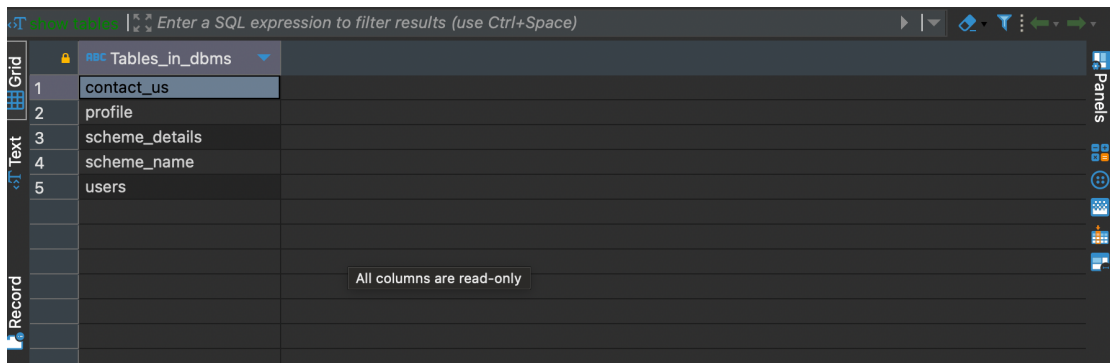
[illegible]

Fig 5.2 all data in table scheme_details



	scheme_id	scheme_name	year
1	1,401	JAN DHAN YOGNA	2,014
2	1,402	MAKE_IN_INDIA	2,014
3	1,403	MISSION SWACHH BHARAT	2,014
4	1,404	BETI BACHAO BETI PADHAO	2,014
5	1,405	Test_Scheme	2,014
6	1,406	Test_Scheme_Running	[NULL]
7	1,407	Test_3	[NULL]
8	1,601	Test_4	[NULL]
9	1,602	Test_5	[NULL]

Fig 5.3 all data in table scheme_name



Tables_in_dbms
contact_us
profile
scheme_details
scheme_name
users

All columns are read-only

Fig 5.4 tables present

Chapter 6

OUTPUT OF WEBPAGES

6.1 Web Interface

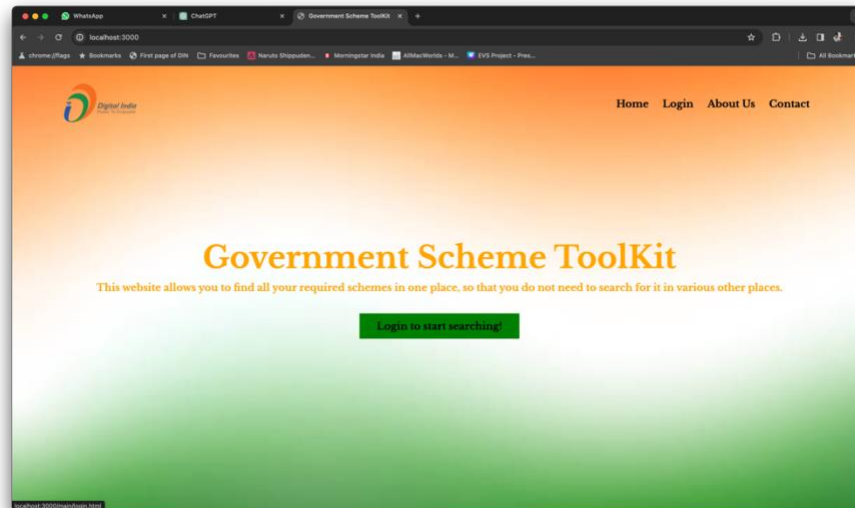


Fig 6.1 Home page

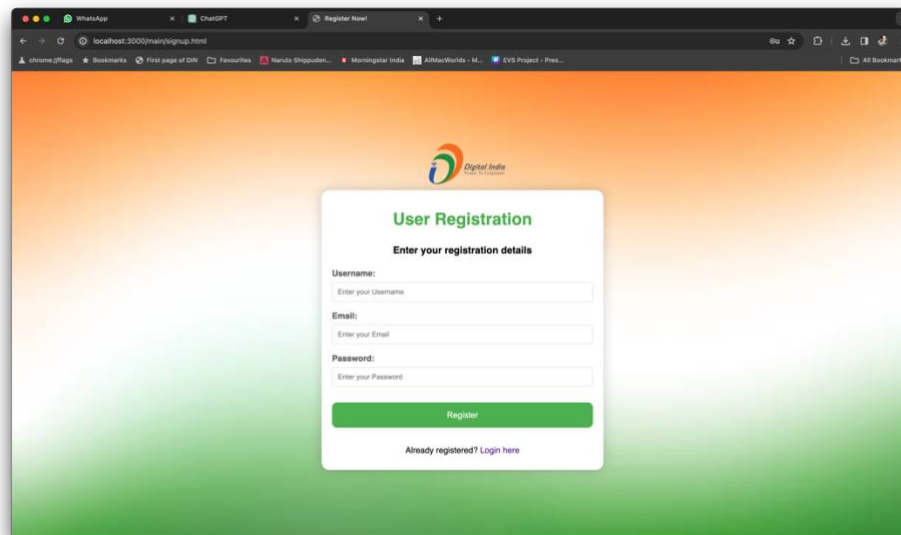


Fig 6.2 User Registration page

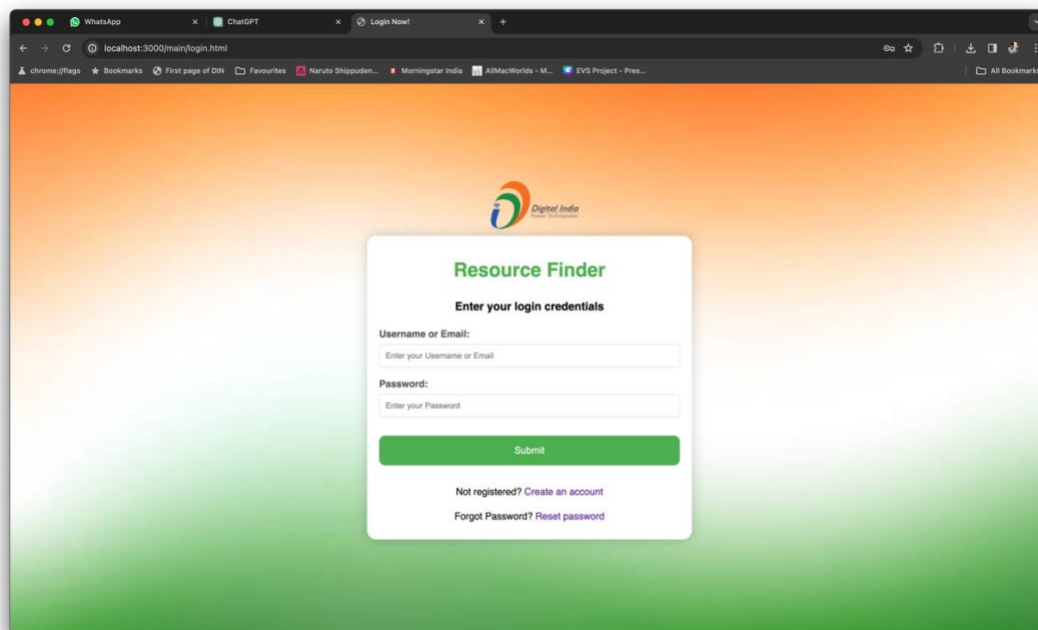


Fig 6.3 Login Page

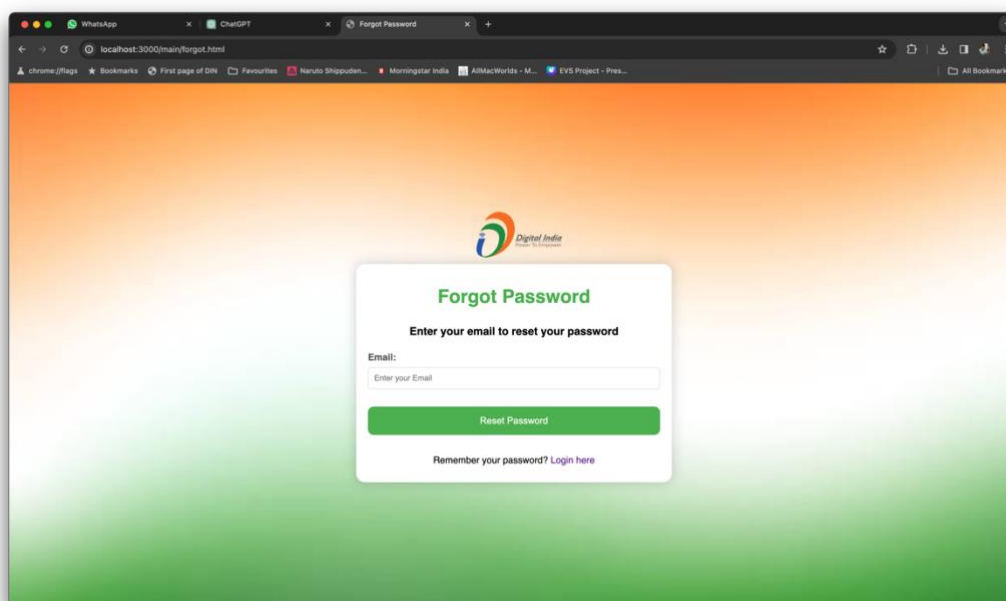


Fig 6.4 Forgot Password Page

Government Schemes Finder

Age: 18

Gender: Male

Category: General

Religion: Hindu

Marital Status: Single

Income: Less than 5 LPA

State: Bihar

Search

Scheme Name	Scheme ID	Age	Gender	Category	Religion	Marital Status	Income	State
JAN DHAN YOGNA	1401	18	Male	General	Hindu	Single	less_than_5	Bihar

Fig 6.5 – User Page

Government Schemes Add - Admin

Scheme ID:

Scheme Name:

Scheme Year:

Age:

Gender: Male

Category: General

Religion:

Marital Status: Single

Income:

Fig 6.6 – Admin Page

Chapter-7

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

In conclusion, the government scheme toolkit represents a significant milestone in the realm of public service innovation, addressing the critical need for centralized access to information about government welfare programs. By consolidating a diverse array of schemes into a user-friendly platform, the toolkit empowers citizens, businesses, and policymakers to navigate and leverage government initiatives more effectively. Through its emphasis on accessibility, transparency, and citizen engagement, the toolkit not only facilitates greater inclusivity and equity but also fosters a culture of active citizenship and accountability.

As we move forward, it is essential to recognize the ongoing importance of the government scheme toolkit in supporting evidence-based policymaking, promoting socio-economic development, and advancing the well-being of society as a whole. Continued efforts to enhance the toolkit's functionality, expand its reach, and strengthen collaboration among stakeholders will be crucial in maximizing its impact and ensuring its sustainability in the long run.

Ultimately, the government scheme toolkit exemplifies the transformative potential of technology in democratizing access to government services and fostering a more responsive and citizen-centric governance model. By harnessing the power of information and connectivity, we can strive towards a future where every individual has equal opportunity to benefit from the resources and support provided by their government, thereby building a more inclusive and prosperous society for generations to come.

Future Enhancements:

Looking ahead, several potential future enhancements can further elevate the effectiveness and utility of the government scheme toolkit:

1. Integration with Government Portals: Enhancing interoperability by integrating the toolkit with existing government portals or platforms can streamline access to scheme information and facilitate seamless interaction between users and government agencies.

2. Personalization and Recommendation Engine: Implementing machine learning algorithms to analyze user behavior and preferences can enable the toolkit to provide personalized recommendations for relevant schemes based on individual profiles and needs.

3. Multi-Language Support: Introducing support for multiple languages can enhance accessibility and cater to diverse linguistic communities, ensuring that all citizens can benefit from the toolkit regardless of their language proficiency.

4. Mobile Application Development: Developing a dedicated mobile application for the toolkit can extend its reach and accessibility, allowing users to access scheme information on-the-go and receive timely notifications about updates and new opportunities.

5. Interactive Data Visualization: Incorporating interactive data visualization tools such as charts, graphs, and maps can enhance the toolkit's ability to present scheme data and trends in a visually engaging and comprehensible manner, facilitating better decision-making and analysis.

6. Community Engagement Features: Implementing features such as discussion forums, user reviews, and social media integration can foster community engagement, enabling users to share experiences, seek advice, and collaborate on initiatives related to government schemes.

7. Real-time Updates and Alerts: Implementing mechanisms to provide real-time updates and alerts about changes to scheme eligibility criteria, application deadlines, and

other relevant information can ensure that users stay informed and up-to-date on the latest developments.

8. Accessibility Enhancements: Incorporating accessibility features such as screen reader compatibility, text resizing options, and keyboard navigation shortcuts can ensure that the toolkit is accessible to users with disabilities, promoting inclusivity and equal access to government services.

9. Partnerships with Non-Governmental Organizations (NGOs): Collaborating with NGOs and community organizations can enhance outreach efforts, improve the dissemination of scheme information, and facilitate last-mile delivery of government benefits to underserved populations.

10. Feedback Mechanisms and User Surveys: Implementing feedback mechanisms, user surveys, and analytics tools can gather valuable insights into user needs, preferences, and satisfaction levels, informing iterative improvements and enhancements to the toolkit's functionality and user experience.

By prioritizing these future enhancements, the government scheme toolkit can evolve into a more robust, user-centric, and impactful platform, driving positive

REFERENCES

[1] User Interface Design:

"Don't Make Me Think" by Steve Krug.

"The Design of Everyday Things" by Don Norman.

Articles from the Nielsen Norman Group (www.nngroup.com).

[2] Data Validation and Integrity:

"Database Management Systems" by Raghu Ramakrishnan and Johannes Gehrke.

"Data Quality: Concepts, Methodologies, and Techniques" by Yang W. Lee.

[3] Access Controls and Security:

"Security Engineering: A Guide to Building Dependable Distributed Systems" by
Ross J. Anderson.

OWASP (Open Web Application Security Project) resources (www.owasp.org).

[4] Efficiency and Process Optimization:

"Business Process Change: A Guide for Business Managers and BPM and Six
Sigma Professionals" by Paul Harmon.

Articles on process optimization from academic journals like the International
Journal of Production Research.