Faculty of Engineering and Technology			
Ramaiah University of Applied Sciences			
Department	Computer Science and Engineering	Programme	B. Tech. in CSE
Semester/Batch	05/2021		
Course Code	20CSC302A	Course Title	Database Systems
Course Leader	Dr. C. Narendra Babu		

Assignment			
Reg. No.	**21ETCS002100**	**Name of Student**	**NISA**

Sections		Marking Scheme	Marks: Max Marks	First Examiner	Moderator
Question 1					
	1	Title and the brief synopsis	**05**		
	2	Method and methodology(data, block diagram, domain, algorithm used, technology used to be mentioned)	**08**		
	3	Implementation	**07**		
	4	Presentation of Results and Report	**05**		
		Total Assignment Marks	**25**		

Course Marks Tabulation

Component	First Examiner	Remarks	Moderator	Remarks
1				
2				
Marks (out of 25)				

Signature of First Examiner **Signature of Moderator**

Instructions to students

1. Maximum marks is **50**.
2. The assignment has to be neatly word processed as per the prescribed format.
3. The printed assignment must be submitted to the course leader.
4. **Submission Date: March 04, 2024**
5. **Submission after the due date is not permitted.**
6. **IMPORTANT**: It is essential that all the sources used in preparation of the assignment must be suitably referenced in the text.
7. Marks will be awarded only to the sections and subsections clearly indicated as per the problem statement/exercise/question
8. Documental evidence for all the components/parts of the assessment such as the reports, photographs, laboratory exam / tool tests are required to be attached to the assignment report in a proper order.

DEADLINE FOR COMPLETION OF THE ASSIGNMENT IS 4TH MARCH 2024

Guide lines for assignment submission:

Contact the course leader for specific application

1. Max two students are allowed. In case if the number is exceeding you need to take the permission explicitly by specifying each roles in the team.
2. Create a problem statement as follows:
    a. Project Title
    b. Brief synopsis: brief write up (about half page or max 1 page) including objectives (Minimum 3 objectives and max 5 objects)
    c. Method and methodology: Mention the domain, apprx number of tables, functionalities etc
    algorithm used, technology used etc. Mention input and output including the dataset used and Block diagram
    d. Team Member names:
        i. Member1 name
        ii. Member 2 name

Your report should be written and submitted as per the following template:

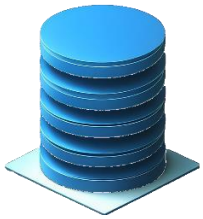
1. Title
2. Synopsis
3. Objectives
4. Methodology/Flow diagram/Block diagram
5. Implementation and result analysis

INDEX

RAMAIAH UNIVERSITY OF APPLIED SCIENCES

TITLE AND BRIEF SYNOPSIS

TITLE and LOGO:

realBase 

Introduction

realBase is based on SQL database management system (DBMS) designed with the user in mind, focusing on flexibility, ease of use, and robust data integrity features.

Users can create numerous tables within a database to store and organize their data logically and efficiently. Each table can represent a different entity or aspect of the application's data model.

It supports all four fundamental CRUD (Create, Read, Update, Delete) operations, enabling users to insert new records, query and retrieve data, update existing records, and delete records from the database.

realBase utilizes SQL (Structured Query Language) as its query language, allowing for the definition, manipulation, and control of data in relational databases.

Objective

The purpose of realBase is multifaceted, focusing on improving the way data is managed, accessed, and utilized across different applications and industries.

It aims to offer a blend of ease of use, robustness, and flexibility, catering to the needs of developers, businesses, and educational institutions in managing their data efficiently and securely.

Scope:

The scope of the realBase project encompasses the following key areas:

- **Functionality:** robust SQL database solution, an intuitive authentication system.
- **Features:** Simplify Data Management, Data Security and Support Complex Operations and Analysis
- **Target Audience**: realBase is targeted towards developers and development teams.

METHOD AND METHODOLOGY

The **development** of realBase involves:

Frontend development:

- HTML (Hypertext Markup Language): Used for structuring the content of web pages as handlebars.
- CSS (Cascading Style Sheets): Used for styling and formatting the layout and appearance of web pages.
- JavaScript: A programming language used for adding interactivity and dynamic behavior to web pages.

Backend development:

- Languages: JavaScript (Node.js)
- Framework: Express.js

Database used: MySQL

We have used input fields in the html as well as format tables to store the data to the database.
Also we used app.get() and app.post() methods to fetch/pass data from one page to another.
We have also executed SQL queries to fetch data from our database that is realBase.

Domain:

- Web Development Domain: enables the creation of robust, data-driven web applications that provide users with the ability to interact with and manipulate data stored in the database effectively. It emphasizes the importance of seamless coordination between frontend and backend components to deliver a cohesive and user-friendly web experience.
- Database Management Domain: Users have the capability to perform essential operations such as inserting, deleting, and updating data in their databases. We also support primary keys and the 'NOT NULL' constraint. We offer SQL database support along with a relational model framework.

- A domain name is the address used to access a website on the internet. It serves as a human-readable label for identifying a specific web location. Domain names are used to represent the unique identity of a website or a web server on the internet.

- Currently we are hosting our website in localhost:3000, if that domain is not present in their respective system it will allow access in their localhost like 8080, 3003 or 3006 etc..

Number of tables used:

- Developers built tables: 2 tables => users and id_tables
- Dynamically created tables whenever users needs to create the table in our realBase.

Technology Used:

- realBase primarily utilizes MySQL for data storage.
- It utilizes technologies related to SQL databases, such as SQL query language i.e., SQL (Structured Query Language).
- Database management systems i.e., MySQL Server.
- The development process involves technologies for interfacing with SQL databases, such as,
  - Node.js for backend development
  - JavaScript for frontend development.

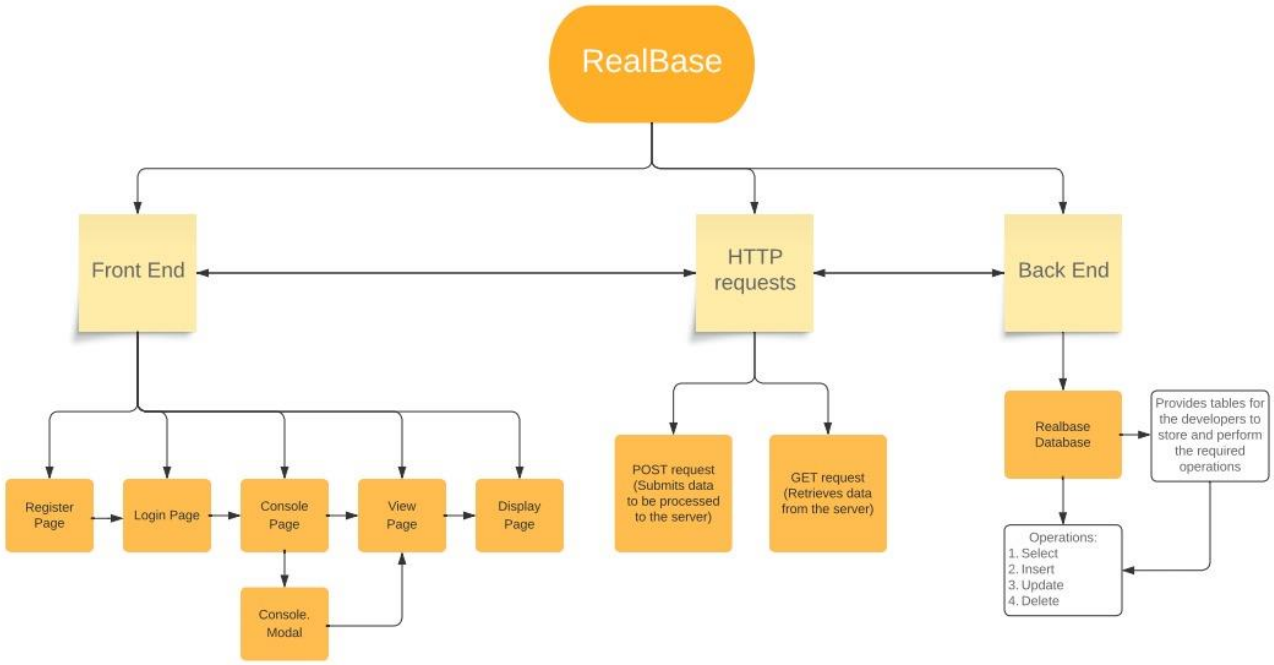


Figure 1.1. Block diagram representing the Working of realBase.

DATA:

Realbase relies solely on SQL databases for data storage and management. It utilizes SQL databases to store user data, application data, and configuration settings. The structured nature of SQL databases allows for efficient querying and retrieval of data, making it suitable for handling various types of information required for Realbase's operation

For example:

User details table in our website:

Result Grid | Filter Rows: | Edit: | Export/Import: | W

id	Name	EmailID	Password
1709957191082	Deeksha	deeksha@gmail.com	$2b$10$07ce6B87YhVIR0rujLuM5e3bKfEqjf/Xa...
1710116229101	Unnathi	unnathi@gmail.com	$2b$10$wqG2VrmZwPJyTAdbFYojKejYBCnnP3z...
1710116263947	Nisa	nisa@gmail.com	$2b$10$VtH/MpROSMkZtUFajLeSr.jHA8fVGFAD...
NULL	NULL	NULL	NULL

A table which contains all the tableNames created by the realBase users

Result Grid | Filter Rows:

ID	Table Name
1709957191082	deeksha
1709957191082	Research_employees
1709957191082	HOD_department
1709957191082	Hod
1709957191082	Students
1709957191082	Universities
1709957191082	Employees_ssn
1709957191082	faculty_list
1709957191082	Course_leaders

Algorithm design:

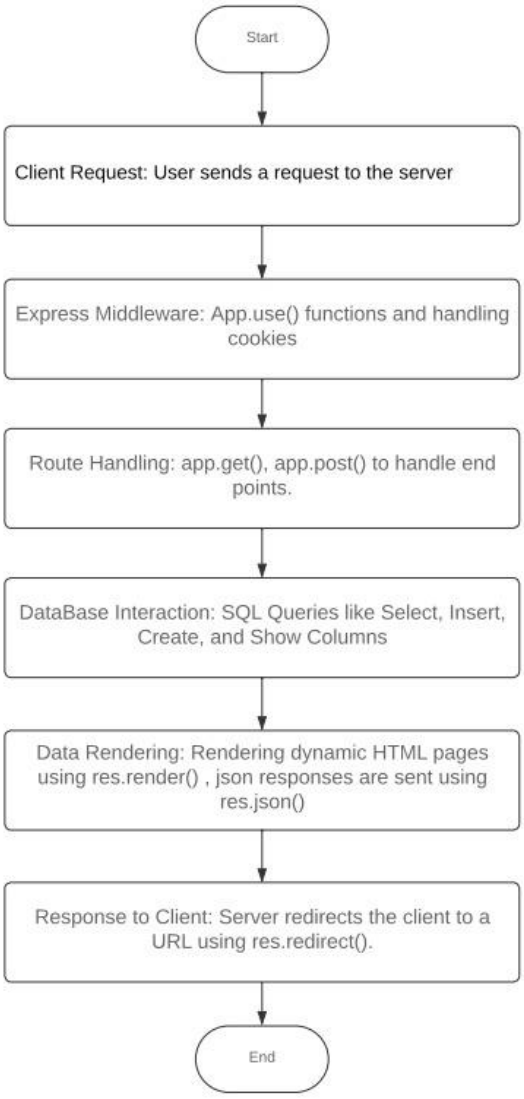


Figure 1.2. Flow chart representing the Working of realBase.

RAMAIAH UNIVERSITY OF APPLIED SCIENCES

IMPLEMENTATION:

app.js

src > app.js > ...

```
const express = require("express");
const hbs = require('hbs');
const app = express();
const path = require("path");
const jwt = require('jsonwebtoken');
const bcrypt = require('bcrypt');
const cookieParser = require('cookie-parser');

const connection = require("../src/database/database");
const static_path = path.join(__dirname, "../views");

app.use(express.json());
app.use(express.urlencoded({extended: true}));
app.set("view engine", "hbs");

app.use(express.static(static_path));
app.use(cookieParser());

const port = process.env.port || 3000;

app.get("/", (req, res) => {
    res.render("index");
});

hbs.registerHelper('firstChar', function(str) {
  return str ? str.charAt(0) : '';
});
```

INSERTING USER DETAILS IN USERS-TABLE:

```
app.post("/register", async (req, res) => {
  try {
      const { id, Name, EmailID, Password } = req.body;
      const hashedPassword = await bcrypt.hash(Password, 10);
      const query = 'INSERT INTO users (id, Name, EmailID, Password) VALUES (?, ?, ?, ?)';

      connection.query(query, [id, Name, EmailID, hashedPassword], (err, results) => {
          if (err) {
              console.error('Database error:', err);
              return res.status(500).send({ message: 'Database error' });
          }
          console.log("User registered successfully");
          res.redirect('/');
      });
  } catch (error) {
      console.error('Error:', error);
      res.status(400).send(error);
  }
});
```

CREATING 2 TABLES

1) TABLE WHICH IS ACCESSED TO THE USER
2) TABLE WHICH WILL STORE ALL THE TABLE NAMES CREATED BY USERS

```
const query = `CREATE TABLE ${nameInput} (${columnDefs});`;
console.log(query);
console.log(connection.query);
connection.query(query, (err, results) =>  {
    if (err) {
        console.error('Could not create table', err);
        return res.status(500).send({ message: 'Could not create table' });
    }
    console.log("Table created successfully");
});

const newTableQuery =  `INSERT INTO id_tables values('${currentID}', '${nameInput}');`;
console.log(`Inserted your ${currentID}, ${nameInput} into our database`);

connection.query(newTableQuery, (err, results) => {
    if (err) {
        return res.status(500);
    }
    res.redirect("/views");
});
```

DISPLAYING COLUMN NAMES OF THE TABLE INTO NEW PAGE:

```
158  app.get("/views", retrieveID, (req, res) => {
159      const currentID = req.ID;
160      const token = req.cookies.token;
161      const decoded = jwt.decode(token);
162      const Name = decoded.Name;
163
164      const newQuery = `SELECT \`Table Name\` FROM id_tables where ID = ${currentID} `;
165      connection.query(newQuery, (err, results) => {
166          if (err) {
167              console.error('Error fetching data from database:', err);
168              res.status(500).send('Internal Server Error');
169              return;
170          }
171          const tableNames = results.map(result => result['Table Name']);
172          res.render('views', { tableNames, Name: Name });
173      });
174  });
```

SELECT * FROM ${currentTable}

```
201  app.get("/display", (req, res) => {
202      const currentTable = req.query.currentTable;
203
204      const dataQuery = `SELECT * FROM ${currentTable}`;
205
206      connection.query(dataQuery, (err, tableData) => {
207          if (err) {
208              console.error('Error fetching data from table:', err);
209              res.status(500).send('Error fetching data from table');
210              return;
211          }
212
213          res.render("display", { selectedTable: currentTable, columns: [], tableData: tableData });
214      });
215  });
```

INSERTING THE VALUES INTO THE TABLES THAT ARE CREATED IN OUR realBase:

```
app.post("/display", (req, res) => {
    const currentTable = req.body.currentTable;
    console.log(currentTable);
    const values = Object.values(req.body);
    const columns = Object.keys(req.body);

    // Prepare the SQL INSERT query
    const indexToRemove = columns.indexOf('currentTable');
    if (indexToRemove > -1) {
        columns.splice(indexToRemove, 1);
        values.splice(indexToRemove, 1);
    }

    const insertQuery = `INSERT INTO ${currentTable} VALUES (${values.map(value => `'${value}'`).join(', ')});`;
    console.log("Insert Query:", insertQuery);

    // Execute the INSERT query
    connection.query(insertQuery, (err, result) => {
        if (err) {
            console.error('Error inserting values into table:', err);
            res.status(500).send('Error inserting values into table');
            return;
        }
        console.log("Values inserted successfully");
        res.redirect("/views"); // Redirect back to the display page
    });
});
```

Application of realBase:

Streamlining backend tasks like user management, and data storage efficiently in some of the following applications,

- e-commerce
- social networking,
- CMS (Content Management System),
- online booking applications etc.,

Advantages of realBase:

1. Simplified Backend Management: realBase reduces complexities in tasks like server setup and authentication.
2. Accelerated Development Timelines: Streamlining processes helps developers deploy applications faster.
3. Scalability and Reliability: realBase's serverless architecture ensures optimal performance and reliability.
4. Cost Efficiency: It eliminates the need for expensive infrastructure maintenance.

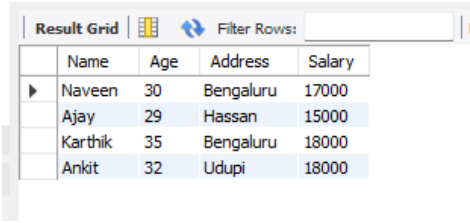
Focus on Innovation:
Developers can concentrate on building innovative features and enhancing user experiences

RESULT AND ANALYSIS:

Results of one random user who creates multiple tables in our realBase and he/she can edit or drop the table too

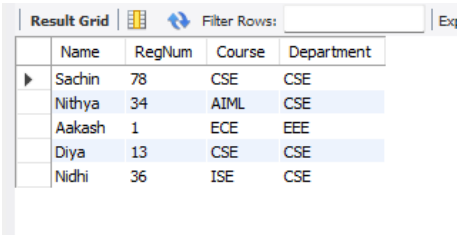
These are the tables which are currently presented in our realBase:

1. Employee table



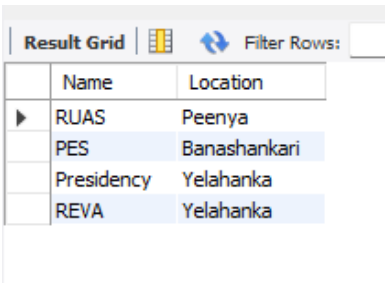
Name	Age	Address	Salary
Naveen	30	Bengaluru	17000
Ajay	29	Hassan	15000
Karthik	35	Bengaluru	18000
Ankit	32	Udupi	18000

2. Students table



Name	RegNum	Course	Department
Sachin	78	CSE	CSE
Nithya	34	AIML	CSE
Aakash	1	ECE	EEE
Diya	13	CSE	CSE
Nidhi	36	ISE	CSE

3. Universities table



Name	Location
RUAS	Peenya
PES	Banashankari
Presidency	Yelahanka
REVA	Yelahanka

---End of Assignment---