MINI PROJECT REPORT ON

"User Management System"

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE AWARD OF THE DEGREE

Bachelor of Engineering

in

Information Technology

Class: T.E

BY

Name :	Roll No.
Ashutosh Mahadik	40
Vaishnavi More	46
Anushka Salve	68
Prathmesh Apsingekar	59
Dhruti Kamat	31

Under the guidance of

Mrs.Kalyani Sonawane

Sinhgad Institutes

DEPARTMENT OF INFORMATION TECHNOLOGY
RMD SINHGAD SCHOOL OF ENGINEERING
WARJE, PUNE-411058

A.Y: 2023- 24

DEPARTMENT OF INFORMATION TECHNOLOGY
RMD SINHGAD SCHOOL OF ENGINEERING
WARJE, PUNE-411058
CERTIFICATE

This is to certify that the Mini Project Report entitled

"User Management System"

Submitted by

Name:	Roll No:
Ashutosh Mahadik	40
Vaishnavi More	46
Anushka Salve	68
Prathmesh Apsingekar	59
Dhruti Kamat	31

is a Bonafide work carried out by him/her under the supervision of Mrs.Kalyani Sonawane and it is submitted towards the partial fulfillment of the requirement for T.E (Information Technology) – 2019 course of Savitribai Phule Pune University, Pune in the academic year 2023-2024.

Mrs.Kalyani Sonawane	Mr.Saurabh Parhad	Dr. V.V. Dixit
Guide	Head	Principal
Dept of IT	Dept of IT	

Place: Pune   Date:

ACKNOWLEDGEMENT

ABSTRACT

In today's interconnected world of digital platforms and online services, effective management of user accounts and their associated data is paramount. A User Management System (UMS) serves as the cornerstone for this endeavor, providing a comprehensive solution for account registration, authentication, authorization, and access control. This abstract explores the vital role of a User Management System in maintaining security, enhancing user experiences, and ensuring regulatory compliance within digital ecosystems. It highlights the versatile applications of UMS across various domains, from social media and e-commerce to corporate intranets. In an era where user data privacy and security are paramount concerns, a robust User Management System is a fundamental component for safeguarding information, streamlining user-related processes, and enabling seamless digital operations.

CONTENT

List of Figures

INTRODUCTION

In the digital age, where online services, websites, and applications are an integral part of our daily lives, managing user accounts and their interactions with these platforms has become a critical task. A User Management System, or UMS, serves as the linchpin of this endeavor, providing a centralized and systematic approach to creating, maintaining, and securing user accounts within an organization's digital ecosystem.A User Management System is a software or infrastructure solution designed to handle a plethora of user-related tasks, ranging from account registration and authentication to authorization and access control. It plays a pivotal role in ensuring the security and efficiency of digital systems, while simultaneously enhancing the user experience.This system allows organizations to not only manage user profiles and permissions but also enforce security measures to protect sensitive information and ensure regulatory compliance. Whether it's a social media platform, an e-commerce website, a corporate intranet, or any online service, a robust User Management System is indispensable for safeguarding data, maintaining user privacy, and simplifying user-related operations.

Problem Statement:

To perform CRUD Operations on the database of user.

Objectives:

The objectives of a User Management System (UMS) encompass a range of crucial functionalities aimed at effectively managing user accounts and maintaining the integrity and security of digital systems. Firstly, a UMS serves to authenticate users, ensuring that individuals accessing a system are indeed who they claim to be. Additionally, it provides a framework for user authorization, controlling access to resources based on assigned roles and permissions, which is essential for data security and user privacy. Furthermore, UMS prioritizes the security of user data, implementing measures to protect sensitive information from unauthorized access or breaches. It enables efficient user account management, allowing for user registration, profile updates, and password management, thereby enhancing the user experience.

MOTIVATION

The motivation for implementing a User Management System (UMS) within an organization or digital platform is multifaceted and grounded in the fundamental principles of security, user experience, efficiency, and regulatory compliance. First and foremost, the imperative of user convenience propels the adoption of UMS. By streamlining account creation, password management, and profile updates, UMS significantly enhances the user experience, fostering greater user satisfaction and engagement.

REQUIREMENT COLLECTION AND ANALYSIS

Requirements collection and analysis is a critical phase in the software development process. It involves gathering, documenting, and thoroughly understanding the needs and expectations of stakeholders to ensure that the software system or application to be developed meets their requirements and objectives. Requirements collection and analysis are essential for the success of a software development project. A clear understanding of what the software needs to achieve and what stakeholders expect is crucial for delivering a system that meets their needs and aligns with the project's objectives. Regular communication and collaboration with stakeholders throughout the process are key to ensuring that requirements are accurately captured and effectively managed.

Collection and their fields.

```
// Use the database

  const db = client.db(dbName);

  // Create a collection for users

  const usersCollection = db.collection('users');

  // Insert a new user

  usersCollection.insertOne({

    username: 'exampleUser',

    password: 'securePassword',

    email: 'user@example.com'

  }, (err, result) => {
```

```
    if (err) throw err;

    console…
```

[11:39 PM, 11/7/2023] Ashutosh Mahadik (RMD): javascript

```javascript
const express = require('express');

const MongoClient = require('mongodb').MongoClient;

const bcrypt = require('bcrypt');

const app = express();

const port = 3000;

// Database Name

const dbName = 'userManagementDB';
```

Hardware & Software Requirements

Hardware Requirements		
Processor	1.9GHz	20 GB Free Space
Ram	1 GB	
Software Requirements		
Operating System	Windows s XP or Any compatible OS	

GRAPHICAL USER INTERFACE

Fig 1: Home Page Of User Information

Fig 2 : Event Page

SOURCECODE

```
*** Website Header - header.ejs

<header class="navbar navbar-dark sticky-top bg-dark flex-md-nowrap p-0 shadow">
  <a class="navbar-brand col-md-3 col-lg-2 me-0 px-3 fs-6" href="/">NodeJs</a>
  <button class="navbar-toggler position-absolute d-md-none collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#sidebarMenu" aria-controls="sidebarMenu" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <form class="nav col-12 col-md-auto flex-fill mb-2 justify-content-center mb-md-0"
 role="search" method="POST" action="/search">
    <input type="search" name="searchTerm" class="form-control form-control-dark w-100
rounded-0 border-0" placeholder="Search..." aria-label="Search">
  </form>
</header>


*** Website Sidebar - sidebar.ejs

<nav id="sidebarMenu" class="col-md-3 col-lg-2 d-md-block bg-light sidebar collapse">
  <div class="position-sticky pt-3 sidebar-sticky">
    <ul class="nav flex-column">
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/">
          <i class="bi bi-house"></i>
          Dashboard
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/about">
          <i class="bi bi-patch-question"></i>
          About
        </a>
      </li>
    </ul>
  </div>
</nav>


*** Dymmy Data
  try {
```
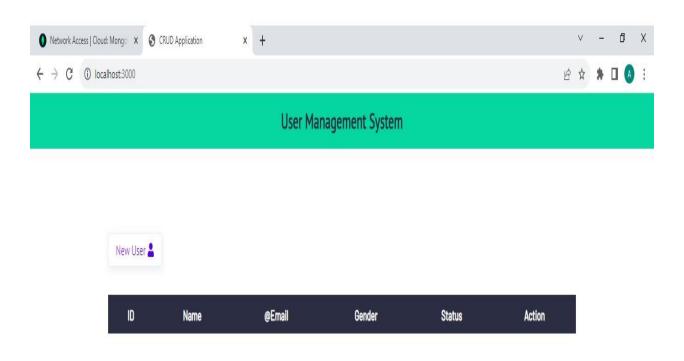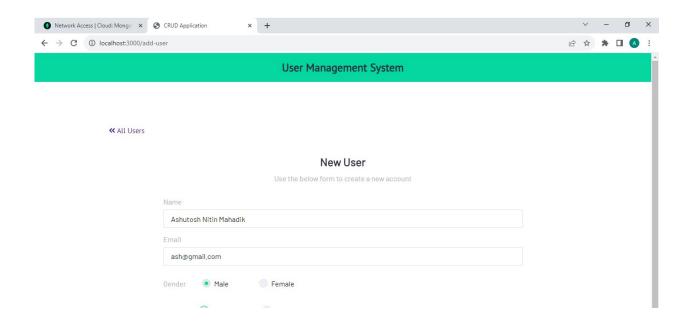
```
*** Dymmy Data
  try {
    await Customer.insertMany([
      {
        firstName: "Raddy",
        lastName: "NodeJs",
        tel: "1-353-218-4881",
        email: "raddy@outlook.couk",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
      },
      {
        firstName: "Aphrodite",
        lastName: "Parker",
        tel: "1-857-407-8574",
        email: "quam@protonmail.net",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
      },
      {
        firstName: "Camden",
        lastName: "Perce",
        tel: "(251) 719-5886",
        email: "aliquam.tincidunt.nunc@icloud.net",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
      },
      {
        firstName: "Emi",
        lastName: "Hutchinson",
        tel: "1-878-674-6876",
        email: "aenean.egestas@aol.org",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
      },
```

```
      },
      {
        firstName: "Chaim",
        lastName: "Holland",
        tel: "1-776-825-8236",
        email: "a@google.couk",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
      },
      {
        firstName: "Harding",
        lastName: "Cameron",
        tel: "1-935-750-3637",
        email: "non.nisi@outlook.edu",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
      },
```

```
        firstName: "Chaim",
        lastName: "Holland",
        tel: "1-776-825-8236",
        email: "a@google.couk",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
    },
    {
        firstName: "Harding",
        lastName: "Cameron",
        tel: "1-935-750-3637",
        email: "non.nisi@outlook.edu",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
    },
    {
        firstName: "Dane",
        lastName: "Kelley",
        tel: "(129) 964-3195",
        email: "morbi@aol.org",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
    },
    {
        firstName: "Emery",
        lastName: "Thornton",
        tel: "(565) 248-4784",
        email: "egestas.blandit.nam@icloud.org",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
    },
    {
        firstName: "Tarik",
        lastName: "Francis",
        tel: "1-679-436-4746",
        email: "lacus@outlook.ca",
        details: "Demo details text.",
```

```
    },
    {
        firstName: "Tarik",
        lastName: "Francis",
        tel: "1-679-436-4746",
        email: "lacus@outlook.ca",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
    },
        {
        firstName: "Rebecca",
        lastName: "Booth",
        tel: "1-548-944-3232",
        email: "sapien@icloud.couk",
        details: "Demo details text.",
        createdAt: Date.now(),
        updatedAt: Date.now()
    },
    {
```

```
{
  firstName: "Solomon",
  lastName: "Larson",
  tel: "(648) 588-4779",
  email: "accumsan.interdum@icloud.net",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
},
{
  firstName: "Tanner",
  lastName: "Morin",
  tel: "(189) 577-5612",
  email: "nec.diam.duis@google.couk",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
},
{
  firstName: "September",
  lastName: "Walton",
  tel: "1-732-422-2492",
  email: "sed.sapien.nunc@icloud.com",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
},
{
  firstName: "Kermit",
  lastName: "Becker",
  tel: "1-163-757-8638",
  email: "id@yahoo.net",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
}
```

```
{
  firstName: "Anish",
  lastName: "Brown",
  tel: "1-163-757-8638",
  email: "Anish@yahoo.net",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
},
{
  firstName: "Duncan",
  lastName: "Woodard",
  tel: "1-163-757-8638",
  email: "Duncan@yahoo.net",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
},
{
  firstName: "Izabella",
  lastName: "Stark",
  tel: "1-163-757-8638",
  email: "Izabella@yahoo.net",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
},
{
  firstName: "Dhruv",
  lastName: "Fields",
  tel: "1-163-757-8638",
  email: "Dhruv@yahoo.net",
  details: "Demo details text.",
  createdAt: Date.now(),
  updatedAt: Date.now()
},
{
```

```
    },
    {
      firstName: "Dhruv",
      lastName: "Fields",
      tel: "1-163-757-8638",
      email: "Dhruv@yahoo.net",
      details: "Demo details text.",
      createdAt: Date.now(),
      updatedAt: Date.now()
    },
    {
      firstName: "Harriet",
      lastName: "Gillespie",
      tel: "1-163-757-8638",
      email: "Harriet@yahoo.net",
      details: "Demo details text.",
      createdAt: Date.now(),
      updatedAt: Date.now()
    },
    {
      firstName: "Chad",
      lastName: "Barton",
      tel: "1-163-757-8638",
      email: "Chad@yahoo.net",
      details: "Demo details text.",
      createdAt: Date.now(),
      updatedAt: Date.now()
    },
    {
      firstName: "Chad",
      lastName: "Barton",
      tel: "1-163-757-8638",
      email: "Chad@yahoo.net",
      details: "Demo details text.",
      createdAt: Date.now(),
      updatedAt: Date.now()
    },
    {
      firstName: "Esmee",
      lastName: "Trujillo",
      tel: "1-163-757-8638",
      email: "Esmee@yahoo.net",
      details: "Demo details text.",
      createdAt: Date.now(),
      updatedAt: Date.now()
    },
  ]);
} catch (error) {
  console.log("err", + error);
}
```

```javascript
const express = require('express');
const MongoClient = require('mongodb').MongoClient;
const bcrypt = require('bcrypt');

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

// Connection URI
const uri = 'mongodb://localhost:27017';

// Database Name
const dbName = 'userManagementDB';

// Middleware to parse JSON requests
app.use(express.json());

// Create a new MongoClient
const client = new MongoClient(uri, { useNewUrlParser: true, useUnifiedTopology: true });

// Connect to the MongoDB server
client.connect((err) => {
  if (err) throw err;

  console.log('Connected to the database');

  // Use the database
  const db = client.db(dbName);

  // Create a collection for users
  const usersCollection = db.collection('users');
```

```
// Connect to the MongoDB server
client.connect((err) => {
  if (err) throw err;

  console.log('Connected to the database');

  // Use the database
  const db = client.db(dbName);

  // Create a collection for users
  const usersCollection = db.collection('users');

  // API endpoint to register a new user
  app.post('/register', async (req, res) => {
    const { username, password, email } = req.body;

    // Hash the password before storing it
    const hashedPassword = await bcrypt.hash(password, 10);

    // Insert the new user into the collection
    const result = await usersCollection.insertOne({
      username,
      password: hashedPassword,
      email
    });

    res.json({ message: 'User registered successfully', user: result.ops[0] });
  });

  // API endpoint to authenticate a user
  app.post('/login', async (req, res) => {
    const { username, password } = req.body;

    // Find the user by username
    const user = await usersCollection.findOne({ username });
```

```
if (user) {
  // Compare the provided password with the hashed password in the database
  const passwordMatch = await bcrypt.compare(password, user.password);

  if (passwordMatch) {
    res.json({ message: 'Login successful' });
  } else {
    res.status(401).json({ message: 'Invalid password' });
  }
} else {
  res.status(404).json({ message: 'User not found' });
}
});

// Start the server
app.listen(port, () => {
  console.log(`Server listening at http://localhost:${port}`);
});
});
```

CONCLUSION

A User Management System plays a pivotal role in modern applications and organizations. Its importance lies in providing robust user authentication, authorization, and administration, which are essential for safeguarding sensitive data and resources. Such systems enhance efficiency by streamlining user account management, ensuring that only authorized individuals gain access to the system. A well-designed User Management System also prioritizes user experience, offering seamless registration, profile management, and password recovery functionalities. Additionally, its scalability, integration capabilities, customization options, and robust audit and logging features make it an indispensable tool for organizations of all sizes. With a focus on data protection and compliance, these systems not only secure user information but also provide valuable insights into user behavior and preferences. In conclusion, a User Management System is the linchpin that ensures security, efficiency, and user satisfaction in today's digital landscape.

REFERENCE

[1] Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 6thEdition, McGraw Hill Publishers, ISBN 0-07-120413-X

[2] Kristina Chodorow,   MongoDB   The   definitive   guide, O'Reilly Publications ISBN:978-93-5110-269- 4,2nd Edition.

[3] Jiawei Han, Micheline Kamber, Jian Pei "Data Mining: concepts and techniques", 2nd Edition, Publisher: Elsevier/Morgan Kaufmann

[4] http://nosql-database.org/.

[5] https://www.tutorialspoint.com