Title: Task Management App

Abstract:

The Task Management App is a full-stack web application designed to help users efficiently manage their tasks. It provides functionalities such as task creation, updates, status tracking, authentication, and real-time collaboration. This documentation provides an in-depth analysis of the app's implementation, including its technical stack, features, pages implemented, and source code for both frontend and backend.

Technical Stack:

To ensure **efficiency, scalability, and security**, HackSync leverages the following technologies:

- **Frontend:** React.js, Tailwind CSS, Vite
- **Backend:** Node.js, Express.js
- **Database:** MongoDB
- **Authentication:** JWT-based Authentication

Features:

- User authentication (Sign up, Login, Logout)

- Create, update, and delete tasks

- Categorize tasks based on priority

- Track task completion status

- Assign tasks to users and track progress

- Role-based access control for admins and users

- Responsive UI for better usability

Pages Implemented:

- Dashboard: Overview of tasks and user progress

- Login & Registration Page: Secure authentication

- Task Management Page: Create and manage tasks

- Profile Page: User settings and customization

- Admin Dashboard: Manage users and tasks

User Roles & Permissions:

1. Admin

Create, update, and delete tasks for all users.
Manage user accounts (add, edit, remove users).
Assign tasks to users.
Monitor overall progress and task completion.
Access all user data and reports.

2. Manager

View and manage tasks assigned to their team members.
Assign tasks to employees.
Edit task details and update statuses.
Generate team reports.

3. Employee/User

Create personal tasks.
View and update assigned tasks.
Change task status (To-Do, In Progress, Completed).
Receive task notifications.

Client side:

Task_Managment_App\client\src\components\CreateTaskModal.jsx"

```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { Button, Modal, Stack } from "react-bootstrap";
import toast from "react-hot-toast";

const CreateTaskModal = ({
  showCreateModal,
  handleCreateModalClose,
  setTasks,
}) => {
  const [title, setTitle] = useState("");
  const [description, setDescription] = useState("");

  const handleCreateTask = async () => {
```

```
  await axios
    .post(
      "http://localhost:4000/api/v1/task/post",
      { title, description },
      {
        withCredentials: true,
        headers: { "Content-Type": "application/json" },
      }
    )
    .then((res) => {
      toast.success(res.data.message);
      setTasks((prevTasks) => [...prevTasks, res.data.task]);
      setTitle("");
      setDescription("");
      handleCreateModalClose();
    })
    .catch((error) => {
      toast.error(error.response.data.message);
    });
};

return (
  <>
    <Modal show={showCreateModal} onHide={handleCreateModalClose}>
      <Modal.Header closeButton>
        <Modal.Title>Create Task</Modal.Title>
      </Modal.Header>
      <Modal.Body>
        <Stack gap={3}>
          <label>Title</label>
          <input
            type="text"
            placeholder="Title"
            value={title}
            onChange={(e) => setTitle(e.target.value)}
          />
        </Stack>
        <br />
        <Stack gap={3}>
          <label>Description</label>
          <input
            type="text"
            placeholder="Description"
            value={description}
            onChange={(e) => setDescription(e.target.value)}
          />
          </Stack>
        </Modal.Body>
        <Modal.Footer>
          <Button variant="secondary" onClick={handleCreateModalClose}>
            Close
          </Button>
          <Button variant="primary" onClick={handleCreateTask}>
            Create
          </Button>
        </Modal.Footer>
      </Modal>
    </>
  );
};

export default CreateTaskModal;
```

Task_Managment_App\client\src\components\Header.jsx

```
import { useState, useEffect } from "react";
import Container from 'react-bootstrap/Container';
import Button from 'react-bootstrap/esm/Button';
import Nav from 'react-bootstrap/Nav';
import Navbar from 'react-bootstrap/Navbar';
import NavDropdown from 'react-bootstrap/NavDropdown';
import { Link, useNavigate } from 'react-router-dom';
import axios from "axios";
import toast from 'react-hot-toast';

function Header({ setTasks, tasks, setIsAuthenticated, isAuthenticated }) {
  const [alltasks, setAllTasks] = useState([]);
  const navigateTo = useNavigate();

  useEffect(() => {
    const fetchTasks = async () => {
      try {
        const { data } = await axios.get(
          "http://localhost:4000/api/v1/task/mytask",
          { withCredentials: true }
        );
        console.log("✅ Fetched Tasks:", data.tasks); // Debugging
        setAllTasks(data.tasks);
        setTasks(data.tasks); // Ensure tasks are set on load
      } catch (error) {
        console.error("❌ Error fetching tasks", error);
      }
    };

    fetchTasks();
  }, [isAuthenticated]);

  const handleLogout = async () => {
    try {
      await axios.get("http://localhost:4000/api/v1/user/logout",
{ withCredentials: true });
      toast.success("Logged out successfully");
      navigateTo("/login");
      setIsAuthenticated(false);
      setTasks([]); // Clear tasks on logout
    } catch (error) {
      toast.error(error.response?.data?.message || "Logout failed");
    }
  };

  const filterTasks = (filterType) => {
    console.log(`□ Filtering tasks by: ${filterType}`);

    if (!alltasks.length) {
      console.warn("⍰ No tasks available to filter");
      return;
    }

    let filteredTasks = [];
    switch (filterType) {
      case "completed":
        filteredTasks = alltasks.filter(task => task.status === "completed");
        break;
      case "incomplete":
        filteredTasks = alltasks.filter(task => task.status === "incomplete");
        break;
      case "archived":
        filteredTasks = alltasks.filter(task => task.archived === true);
        break;
      case "all":
      default:
        filteredTasks = alltasks;
        break;
    }

    console.log("□ Filtered Tasks:", filteredTasks); // Debugging
    setTasks([...filteredTasks]); // Force UI update
  };

  return (
    <Navbar expand="lg" className={`bg-body-tertiary ${!isAuthenticated ?
"d-none" : ""}`}>
      <Container>
        <Navbar.Brand href="/">TASK MANAGEMENT</Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="me-auto">
            <Link to="/" className='text-decoration-none d-flex align-items-
center link-light'>
              Home
            </Link>
            <NavDropdown title="Filter Tasks" id="basic-nav-dropdown">
              <NavDropdown.Item onClick={() => filterTasks("all")}>All
Tasks</NavDropdown.Item>
              <NavDropdown.Item onClick={() =>
filterTasks("completed")}>Completed Tasks</NavDropdown.Item>
              <NavDropdown.Item onClick={() =>
filterTasks("incomplete")}>Incomplete Tasks</NavDropdown.Item>
              <NavDropdown.Item onClick={() =>
filterTasks("archived")}>Archived Tasks</NavDropdown.Item>
            </NavDropdown>
            <Link to="/profile" className='text-decoration-none d-flex align-
items-center link-light'>
              Profile
            </Link>
            <Button className="bg-transparent border-0"
onClick={handleLogout}>
              LOGOUT
            </Button>
          </Nav>
        </Navbar.Collapse>
      </Container>
    </Navbar>
  );
}

export default Header;
```

Task_Managment_App\client\src\components\Home.jsx"

```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { Modal, Button, Card, Stack } from "react-bootstrap";
import toast from "react-hot-toast";
import CreateTaskModal from "./CreateTaskModal";
import UpdateTaskModal from "./UpdateTaskModal";
import ViewTaskModal from "./ViewTaskModal";
import { FaEye } from "react-icons/fa";
import { MdEdit, MdDelete } from "react-icons/md";
import { Navigate } from "react-router-dom";

const Home = ({ isAuthenticated, tasks, setTasks, taskTitle }) => {
  const [showCreateModal, setShowCreateModal] = useState(false);
  const [showUpdateModal, setShowUpdateModal] = useState(false);
  const [showViewModal, setShowViewModal] = useState(false);
  const [viewTaskId, setViewTaskId] = useState(null);
  const [updatedTaskId, setUpdateTaskId] = useState(null);

  useEffect(() => {
    console.log("□ Updated Tasks in Home.jsx:", tasks);
  }, [tasks]); // Debugging: Check if tasks update when filters are applied

  const deleteTask = async (id) => {
    try {
      const { data } = await axios.delete(
        `http://localhost:4000/api/v1/task/delete/${id}`,
        { withCredentials: true }
      );
      toast.success(data.message);

      // Remove the deleted task from UI
      setTasks((prevTasks) => prevTasks.filter((task) => task._id !== id));
    } catch (error) {
      toast.error(error.response?.data?.message || "Failed to delete task");
    }
  };

  const handleCreateModalClose = () => setShowCreateModal(false);
  const handleUpdateModalClose = () => setShowUpdateModal(false);
  const handleViewModalClose = () => setShowViewModal(false);
  const handleCreateModalShow = () => setShowCreateModal(true);

  const handleUpdateModalShow = (id) => {
    setUpdateTaskId(id);
    setShowUpdateModal(true);
  };

  const handleViewModalShow = (id) => {
    setViewTaskId(id);
    setShowViewModal(true);
  };

  if (!isAuthenticated) {
    return <Navigate to={"/login"} />;
  }

  return (
    <div className="container my-4">
      <div className="row mb-3">
        <div className="col">
          <h1>{taskTitle || "Tasks"}</h1>
        </div>
        <div className="col text-end">
          <Button variant="primary" onClick={handleCreateModalShow}>
            Create Task
          </Button>
        </div>
      </div>

      <div className="row">
        {tasks && tasks.length > 0 ? (
          tasks.map((task) => (
            <div key={task._id} className="col-lg-3 col-md-4 col-sm-6">
              <Card style={{ marginBottom: "20px", minHeight: "400px" }}>
                <Card.Body className="d-flex justify-content-between flex-column">
                  <Stack gap={2}>
                    <Card.Title className="mb-2" style={{ height: "50px" }}>
                      {task && task.title.length <= 40
                        ? task.title
                        : task.title.slice(0, 40) + "..."}
                    </Card.Title>
                    <Card.Text>
                      {task && task.description.length <= 300
                        ? task.description
                        : task.description.slice(0, 300) + "..."}
                    </Card.Text>
                  </Stack>
                  <Stack
                    direction="horizontal"
                    className="justify-content-end"
                    gap={2}
                  >
                    <MdEdit
                      onClick={() => handleUpdateModalShow(task._id)}
                      className="fs-3 cursor-pointer"
                    />
                    <MdDelete
                      onClick={() => deleteTask(task._id)}
                      className="fs-3 cursor-pointer text-danger"
                    />
                    <FaEye
                      onClick={() => handleViewModalShow(task._id)}
                      className="fs-3 cursor-pointer"
                    />
                  </Stack>
                </Card.Body>
              </Card>
            </div>
          ))
        ) : (
          <h3 className="text-center mt-4 text-muted">
            No tasks available for {taskTitle}
          </h3>
        )}
      </div>

      <CreateTaskModal
        handleCreateModalClose={handleCreateModalClose}
        showCreateModal={showCreateModal}
        setTasks={setTasks}
      />

      <UpdateTaskModal
        handleUpdateModalClose={handleUpdateModalClose}
        showUpdateModal={showUpdateModal}
        id={updatedTaskId}
        setTasks={setTasks}
      />

      <ViewTaskModal
        handleViewModalClose={handleViewModalClose}
        showViewModal={showViewModal}
        id={viewTaskId}
      />
    </div>
  );
};

export default Home;
```

```
Task_Managment_App\client\src\components\Login.jsx"
import Button from "react-bootstrap/Button";
import Form from "react-bootstrap/Form";
import toast from "react-hot-toast";
import axios from "axios";
import { useState } from "react";
import { Container } from "react-bootstrap";
import { Link, Navigate } from "react-router-dom";

function Login({ isAuthenticated, setIsAuthenticated }) {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleLogin = async (e) => {
    e.preventDefault();
    axios
      .post(
        "http://localhost:4000/api/v1/user/login",
        { email, password },
        {
          withCredentials: true,
          headers: { "Content-Type": "application/json" },
        }
      )
      .then((res) => {
        setEmail("");
        setPassword("");
        setIsAuthenticated(true);
        toast.success(res.data.message);
      })
      .catch((error) => {
        console.log(error);
        toast.error(error.response.data.message);
      });
  };
  if (isAuthenticated) {
    return <Navigate to={"/"} />;
  }
  return (
    <Container
      className="d-flex justify-content-center align-items-center overflow-y-hidden"
      style={{ minHeight: "800px" }}
    >
      <Form onSubmit={handleLogin} className="w-100">
        <h3 className="text-center ">LOGIN</h3>
        <Form.Group className="mb-3" controlId="formBasicEmail">
          <Form.Label>Email address</Form.Label>
          <Form.Control
            type="email"
            placeholder="Enter email"
            value={email}
            onChange={(e) => setEmail(e.target.value)}
          />
          <Form.Text className="text-muted">
            We'll never share your email with anyone else.
          </Form.Text>
        </Form.Group>

        <Form.Group className="mb-3" controlId="formBasicPassword">
          <Form.Label>Password</Form.Label>
          <Form.Control
            type="password"
            placeholder="Password"
            value={password}
            onChange={(e) => setPassword(e.target.value)}
          />
        </Form.Group>
        <Form.Group className="text-center">
          <Form.Label>
            Not Registered?{" "}
            <Link to={"/register"} className="text-decoration-none ">
              REGISTER NOW
            </Link>
          </Form.Label>
        </Form.Group>
        <Button
          variant="warning"
          type="submit"
          className="w-100 text-light fw-bold fs-5"
        >
          Submit
        </Button>
      </Form>
    </Container>
  );
}

export default Login;
```

Task_Managment_App\client\src\components\Register.jsx"

```
import axios from "axios";
import React, { useState } from "react";
import toast from "react-hot-toast";
import Button from "react-bootstrap/Button";
import Form from "react-bootstrap/Form";
import { Container } from "react-bootstrap";
import { Link, Navigate } from "react-router-dom";

const Register = (setIsAuthenticated,isAuthenticated) => {
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [phone, setPhone] = useState("");
  const [avatar, setAvatar] = useState("");

  const avatarHandler = (e) => {
    const file = e.target.files[0];
    setAvatar(file);
  };
  const handleRegister = async (e) => {
    e.preventDefault();
    const formData = new FormData();
    formData.append("name", name);
    formData.append("email", email);
    formData.append("phone", phone);
    formData.append("password", password);
    formData.append("avatar", avatar);

    await axios
    .post("http://localhost:4000/api/v1/user/register", formData, {
      withCredentials: true,
      headers: { "Content-Type": "multipart/form-data" },
    })
    .then((res) => {
      setName("");
      setEmail("");
      setPhone("");
      setPassword("");
      setAvatar("");
      setIsAuthenticated(true);
      toast.success(res.data.message);
    })
    .catch((error) => {
      toast.error(error.response.data.message);
    });
  };
```

```
      if (isAuthenticated) {
        return <Navigate to={"/"} />;
      }
    return  <Container
    className="d-flex justify-content-center align-items-center"
    style={{ minHeight: "800px" }}
  >
    <Form onSubmit={handleRegister} className="w-100">
      <h3 className="text-center ">REGISTER</h3>
      <Form.Group className="mb-3" controlId="formBasicEmail">
        <Form.Label>Name</Form.Label>
        <Form.Control
          type="text"
          placeholder="Enter Your Name"
          value={name}
          onChange={(e) => setName(e.target.value)}
        />
      </Form.Group>

      <Form.Group className="mb-3" controlId="formBasicPassword">
        <Form.Label>Email</Form.Label>
        <Form.Control
          type="email"
          placeholder="Enter Your Email"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
        />
      </Form.Group>
      <Form.Group className="mb-3" controlId="formBasicPassword">
        <Form.Label>Phone Number</Form.Label>
        <Form.Control
          type="number"
          placeholder="Your Phone Number"
          value={phone}
          onChange={(e) => setPhone(e.target.value)}
        />
      </Form.Group>
      <Form.Group className="mb-3" controlId="formBasicPassword">
        <Form.Label>Password</Form.Label>
        <Form.Control
          type="password"
          placeholder="Password"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
        />
    </Form.Group>
    <Form.Group className="mb-3" controlId="formBasicPassword">
      <Form.Label>Avatar</Form.Label>
      <Form.Control type="file" onChange={avatarHandler} />
    </Form.Group>
    <Form.Group className="text-center">
      <Form.Label>
        Already Registered?{" "}
        <Link to={"/login"} className="text-decoration-none ">
          LOGIN
        </Link>
      </Form.Label>
    </Form.Group>
    <Button
      variant="warning"
      type="submit"
      className="w-100 text-light fw-bold fs-5"
    >
      Submit
    </Button>
  </Form>
</Container>
};

export default Register
```

Server side:

App.jsx

```
import express from 'express';
import dotenv from 'dotenv';
import cookieParser from 'cookie-parser';
import cors from 'cors';
import {dbConnection} from './database/dbConnection.js';
import fileUpload from 'express-fileupload';
import {errorMiddleware} from './middlewares/error.js';
import userRouter from './routes/userRouter.js';
import taskRouter from './routes/taskRouter.js';

const app = express();
dotenv.config({path: './config/config.env'});

app.use(
   cors({
   origin: [process.env.FRONTEND_URL],
   methods: ["GET","PUT","DELETE","POST"],
   credentials: true
   })
);

app.use(cookieParser());
app.use(express.json());
app.use(express.urlencoded({extended: true}));

app.use(
   fileUpload({
   useTempFiles: true,
   tempFileDir: '/tmp/',
   })
);

app.use('/api/v1/user', userRouter);
app.use('/api/v1/task', taskRouter);

dbConnection();

app.use(errorMiddleware);

export default app;
```

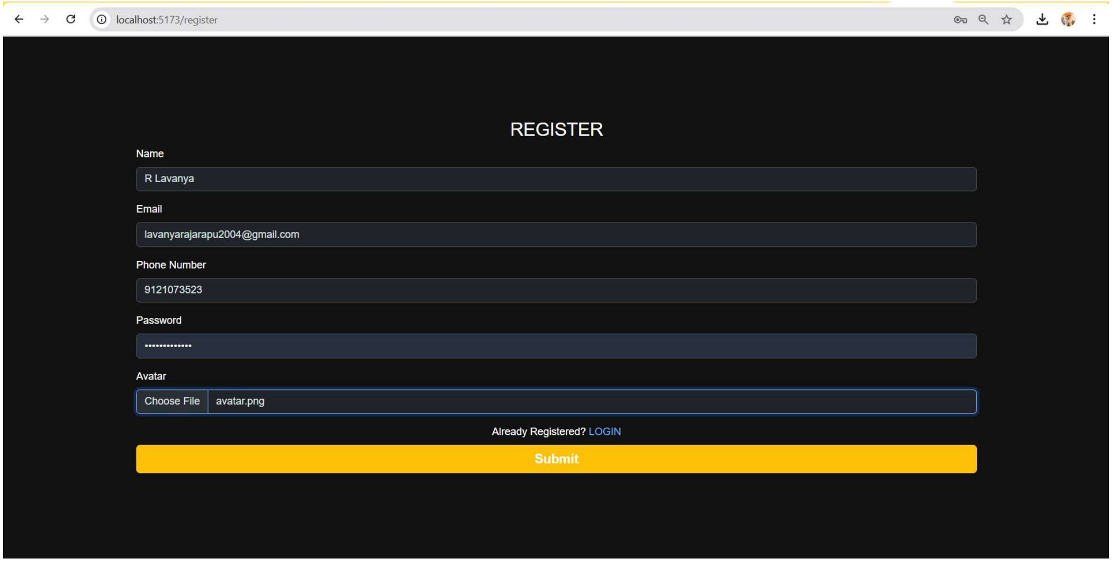
Server.js

```
import app from './app.js';
import cloudinary from 'cloudinary';

cloudinary.v2.config({
   cloud_name: process.env.CLOUDINARY_CLIENT_NAME,
   api_key: process.env.CLOUDINARY_CLIENT_API,
   api_secret: process.env.CLOUDINARY_CLIENT_SECRET,
})

app.listen(process.env.PORT, () => {
   console.log(`Server listening on port: ${process.env.PORT}`);
});
```

Outputs:



localhost:5173/register

REGISTER

Name

R Lavanya

Email

lavanyarajarapu2004@gmail.com

Phone Number

9121073523

Password

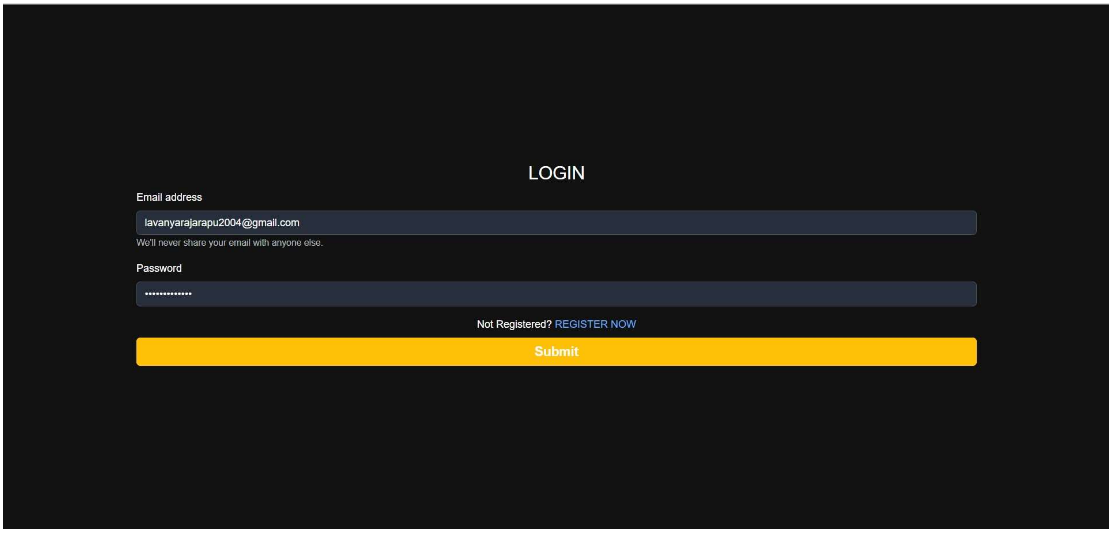
••••••••••••••

Avatar

Choose File avatar.png

Already Registered? LOGIN

Submit



LOGIN

Email address

lavanyarajarapu2004@gmail.com

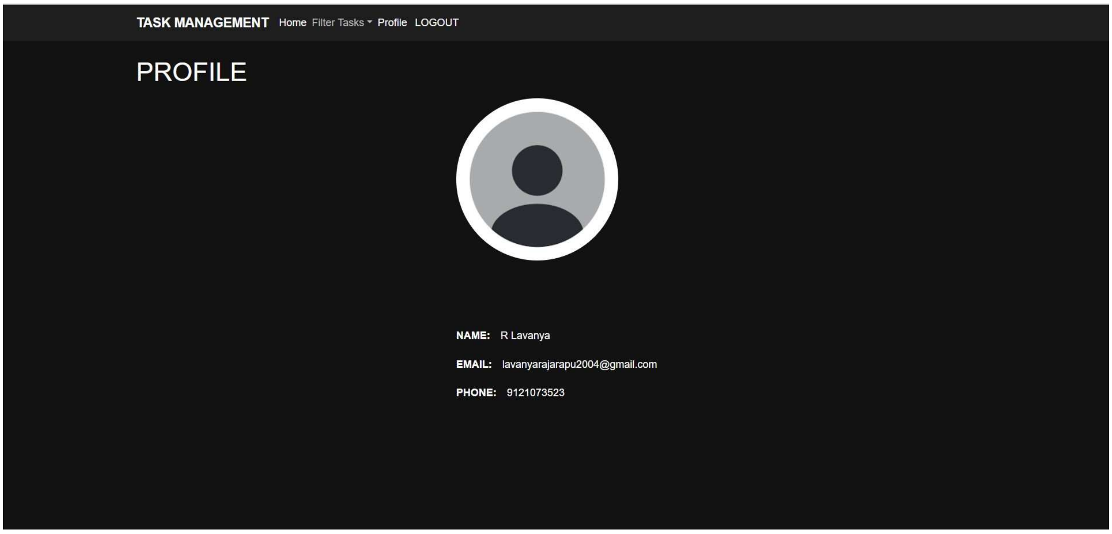
We'll never share your email with anyone else.

Password

••••••••••••••

Not Registered? REGISTER NOW

Submit



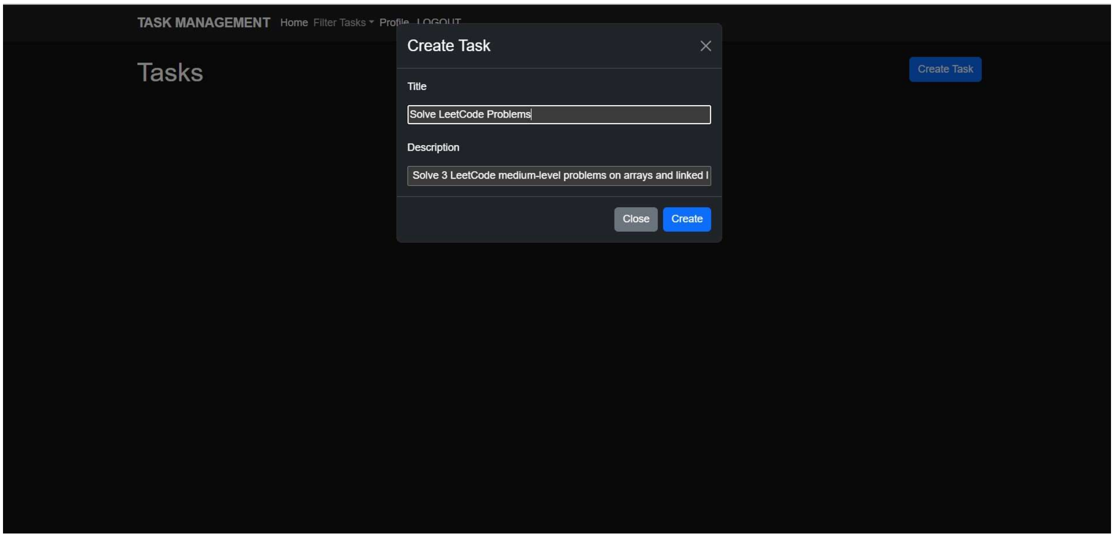
TASK MANAGEMENT Home Filter Tasks Profile LOGOUT

PROFILE

NAME: R Lavanya

EMAIL: lavanyarajarapu2004@gmail.com

PHONE: 9121073523



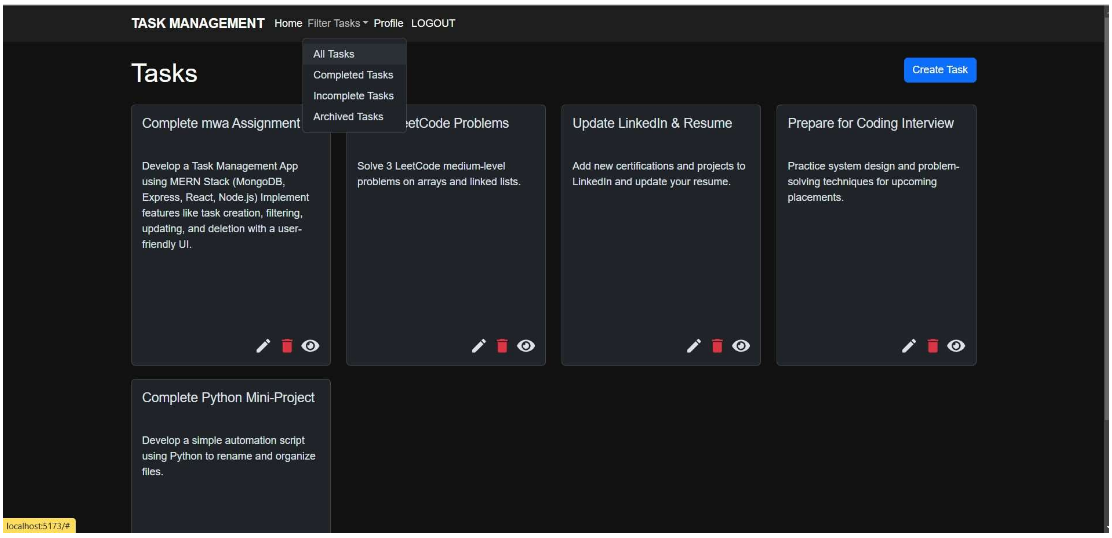
Create Task

Title

Solve LeetCode Problems

Description

Solve 3 LeetCode medium-level problems on arrays and linked l



Complete mwa Assignment

Develop a Task Management App using MERN Stack (MongoDB, Express, React, Node.js) Implement features like task creation, filtering, updating, and deletion with a user-friendly UI.

etCode Problems

Solve 3 LeetCode medium-level problems on arrays and linked lists.

Update LinkedIn & Resume

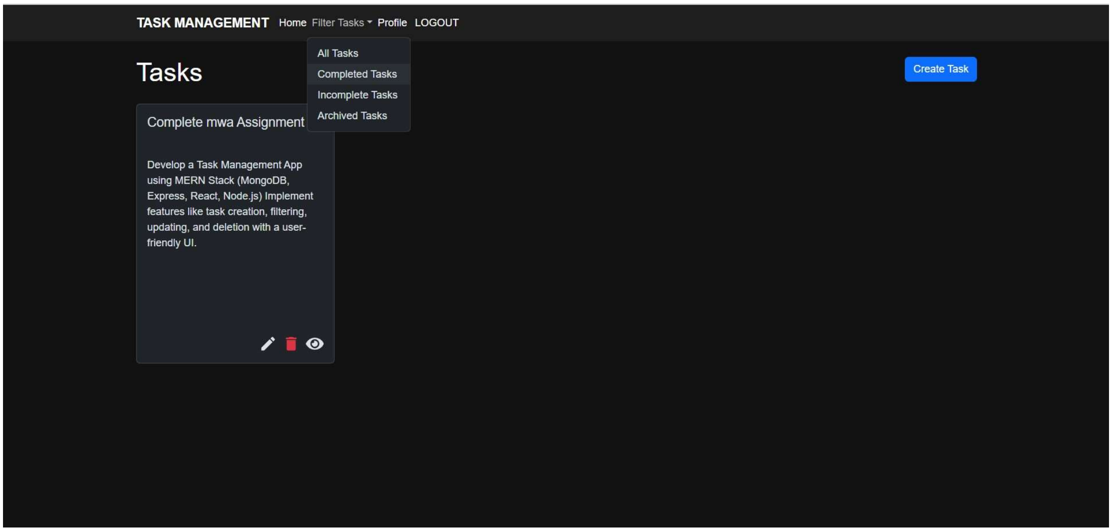
Add new certifications and projects to LinkedIn and update your resume.

Prepare for Coding Interview

Practice system design and problem-solving techniques for upcoming placements.

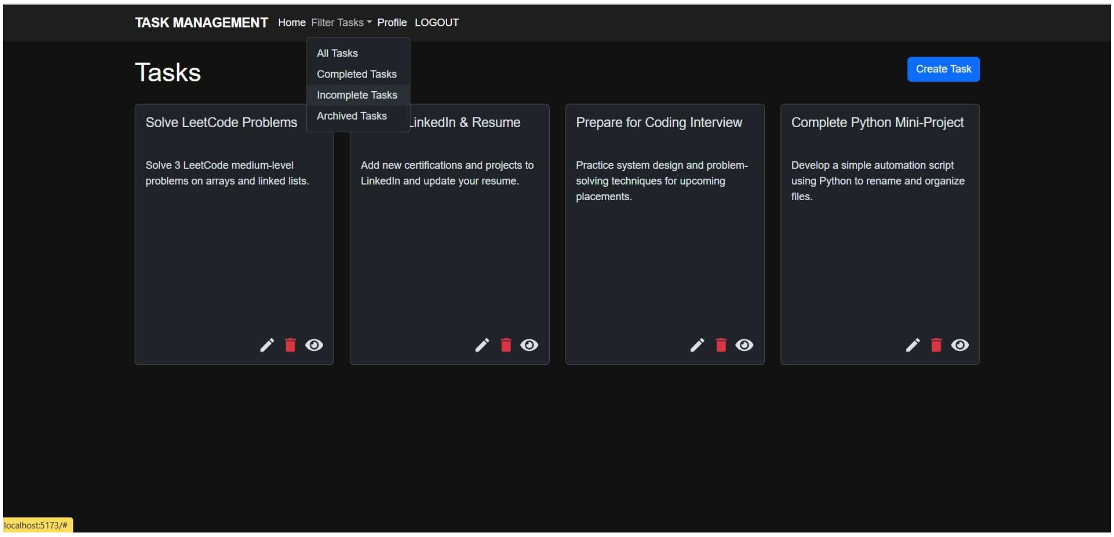
Complete Python Mini-Project

Develop a simple automation script using Python to rename and organize files.



Complete mwa Assignment

Develop a Task Management App using MERN Stack (MongoDB, Express, React, Node.js) Implement features like task creation, filtering, updating, and deletion with a user-friendly UI.



Tasks

Solve LeetCode Problems

Solve 3 LeetCode medium-level problems on arrays and linked lists.

LinkedIn & Resume

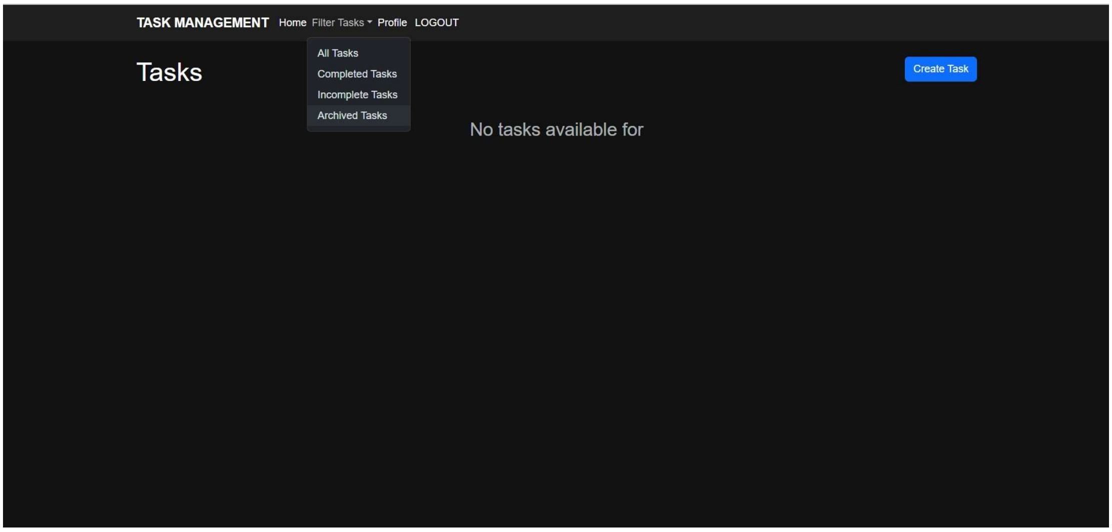
Add new certifications and projects to LinkedIn and update your resume.

Prepare for Coding Interview

Practice system design and problem-solving techniques for upcoming placements.

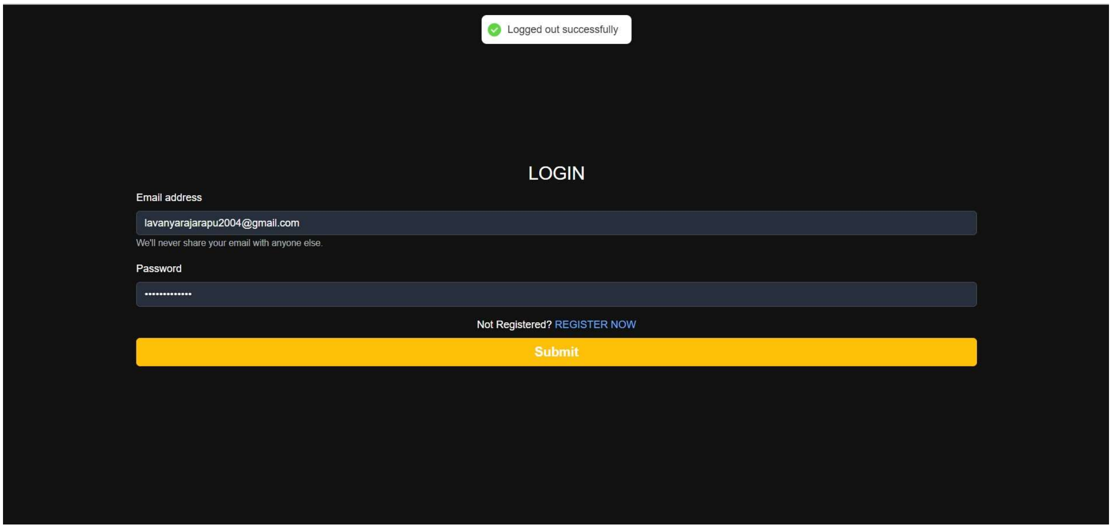
Complete Python Mini-Project

Develop a simple automation script using Python to rename and organize files.



Tasks

No tasks available for



Logged out successfully

LOGIN

Email address

lavanyarajarapu2004@gmail.com

We'll never share your email with anyone else.

Password

••••••••••••••

Not Registered? REGISTER NOW