

A

**Project Report On
HR Management System
Submitted to**



**Great Ganges Institute of Technology, Unnao
(AFFILIATED TO CSJMU UNIVERSITY, KANPUR)**

**In partial fulfilment of the requirements of the award of the degree of
BACHELOR OF COMPUTER APPLICATION (BCA)**

**UNDER THE GUIDANCE
Of**

Mr. Kumar Ajay Savita

Submitted by:

Shivam Prajapati

Roll No: 22015003491

Session: 2024-25

Project Index

| <u>Sr.no</u> | <u>Content</u> | <u>Page No</u> |
|--------------|---|----------------|
| 1 | Acknowledgment | 3 |
| 2 | Certificate | 4 |
| 3 | Introduction | 5 |
| | 4.1 Motivation and overview | |
| | 4.2 Objective | |
| 5 | Software Requirements and Analysis | 7-12 |
| | 5.1 Define the problem | |
| | 5.2 Define the module and their functionalities (SRS) | |
| 6 | Software Design | 13-25 |
| | 6.1 DFD | |
| | 6.2 ER Diagram | |
| | 6.3 Database Table | |
| 7 | Implementation and User Interface | 26-94 |
| | 7.1 Coding | |
| | 7.2 Output | |
| 8 | Conclusion | 95 |
| 9 | References | 96 |



Great Ganges Institute of Technology, Unnao
Address: 22 Km Milestone, Kanpur-Lucknow Highway,
P.o. Singrausi, Gadan Khera, Uttar Pradesh- 209801

ACKNOWLEDGEMENT

I want to convey my heartfelt appreciation and gratitude to my professors as well as my supervisors for providing me with the chance to work on this wonderful project “HR Management System”, which also aided me in conducting extensive study and learning about a bunch of new topics. I am quite grateful to them.

Second, I want to thank my family and friends for their assistance in completing this project in such a period of time. It was quite useful in terms of expanding my knowledge and abilities. Thank you to everyone who has helped completing the Project.



Great Ganges Institute of Technology, Unnao
Address: 22 Km Milestone, Kanpur-Lucknow Highway,
P.o. Singrausi, Gadan Khera, Uttar Pradesh- 209801

CERTIFICATE

This is to certify that **Mr. Shivam Prajapati**, Roll Number **22015003491**, a student of **BCA** at **Great Ganges Institute of Technology, Unnao**, has successfully carried out the project work presented in his report titled **“HR Management System”** for the partial fulfillment of the degree of **Bachelor of Computer Application (B.C.A)** from **C.S.J.M. Kanpur University**, for the academic session **2024-25**.

Under the Guidance of
Mr. Ajay Savita

Principal
Great Ganges Institute of Technology, Unnao



Great Ganges Institute of Technology, Unnao
Address: 22 Km Milestone, Kanpur-Lucknow Highway,
P.o. Singrausi, Gadan Khera, Uttar Pradesh- 209801

ABSTRACT

The **HR Management System** is a web-based application developed to simplify and automate the processes of **employee management** and **leave management** within an organization. The primary objective of this system is to help the HR department maintain accurate employee records and efficiently handle leave requests in a digital format, eliminating the need for manual tracking and reducing the chances of human error.

This system allows HR personnel to easily add, update, and manage employee details such as personal information, job roles, and contact details. Additionally, it provides a structured workflow for managing leave applications, including applying, approving, and tracking leave status, ensuring transparency and accountability.

Designed with a user-friendly interface, the system can be used by non-technical users with ease. By digitizing the core HR processes, the application saves time, enhances productivity, and ensures data accuracy. The system offers quick access to employee and leave records whenever needed, supporting better decision-making and effective workforce management.

This solution is especially beneficial for organizations looking to transition towards a digital work environment and improve the efficiency of their HR operations.

Chapter -1

INTRODUCTION

Basic Introduction:

The HR Management System is a simple and efficient web-based application developed to manage key human resource functions such as employee records and leave management. In many organizations, handling employee information and leave requests manually can be time-consuming and prone to errors. This system is designed to overcome these challenges by providing a digital solution that is easy to use and accessible to non-technical users.

The system allows HR personnel to maintain detailed employee profiles, track attendance, and manage leave applications in a streamlined way. By reducing paperwork and automating routine tasks, the application helps improve overall efficiency and accuracy within the HR department.

This project is especially useful for small to mid-sized organizations that want to shift from manual processes to a more organized and digital mode of working. The system is user-friendly, time-saving, and helps in maintaining transparency and proper documentation of employee-related activities.

Objective:

The main objective of the HR Management System is to develop a user-friendly web application that simplifies the management of employee information and leave records. The system aims to:

- Digitally store and manage employee details.
- Provide an efficient way to apply, approve, and track employee leaves.
- Eliminate manual errors in maintaining HR records.
- Enable quick access to employee and leave data when required.
- Save time and improve the productivity of the HR department.
- Support non-technical users through an easy-to-use interface.

Scope:

The scope of this project is limited to two main HR functionalities: Employee Management and Leave Management. The system provides the following features:

- Add, update, and delete employee details.
- View and manage a list of all employees.
- Allow employees to apply for leave.
- Enable HR/admin to approve or reject leave applications.
- Track leave history and status.
- Secure login system for authorized access.
- Simple and responsive design for ease of use on multiple devices.

Tools and Technology:

| Category | Tools / Technology | Purpose |
|--------------------|---------------------|---|
| Frontend | React.js | Building interactive user interfaces |
| Backend | Node.js, Express.js | Web application framework for Node.js |
| Database | MongoDB | NoSQL database to store employee and leave data |
| API Testing | Postman | Testing and debugging APIs |
| Styling (optional) | CSS / Bootstrap | Styling and responsive design |

Chapter -2

System analysis

Preliminary analysis & information gathering:

Before starting the development of the HR Management System, a preliminary analysis was conducted to understand the requirements, challenges, and expectations of the users. The main focus was to simplify the management of employee data and leave tracking through a digital solution.

1. Problem Identification:

In many organizations, HR-related tasks like maintaining employee records and managing leave requests are handled manually or through basic spreadsheets. This often leads to:

- Time-consuming processes
- Human errors
- Mismanagement of records
- Lack of centralized access to information

2. Information Gathering Techniques:

| Method | Purpose |
|-----------------|--|
| Observation | Analyzing how employee and leave data is currently managed |
| Document Review | Studying existing formats (manual registers, Excel sheets) used for HR work. |

3. Key Requirements Identified:

From the analysis, the following functional requirements were identified:

- A centralized system to manage all employee information.
- A module for applying and approving leave requests.
- A user-friendly interface for both admin and employees.
- Proper access control and authentication.
- Real-time status updates for leaves applications.
- Easy data retrieval and reports.

Feasibility study

The feasibility study is conducted to evaluate whether the HR Management System is practical and beneficial for the organization. It helps determine whether the project is viable in terms of technical, operational, and economic aspects.

Technical Feasibility:

- The system is being developed using the **MERN Stack (MongoDB, Express.js, React.js, Node.js)**, which is widely used, well-supported, and reliable for building scalable web applications.
- All required technologies are open-source and freely available, minimizing technical constraints.
- The development tools (e.g., VS Code, Postman) are already accessible and well-suited for the project.
- The existing infrastructure (modern browsers, internet access) supports the implementation of a web-based system.

Operational Feasibility:

- The system is designed to be user-friendly so that even non-technical users (e.g., HR staff) can easily manage employee and leave data.
- It reduces manual effort, improves accuracy, and speeds up routine HR tasks.
- With proper training, users can quickly adapt to the system.
- Data can be accessed and managed easily from anywhere with internet access, enhancing flexibility.

Economic Feasibility:

- Since all core technologies used are open-source, there are no major costs involved in purchasing software or licenses.
- Development and deployment can be done with minimal resources.
- The system will help reduce manual labour, printing, and storage costs over time, resulting in long-term savings.

System Requirement specification

The System Requirement Specification defines the functional and non-functional requirements of the **HR Management System**.

Functional Requirements:

| Module | Description |
|---------------------|--|
| User Authentication | Users (Admin/HR) must log in with valid credentials to access the system. |
| Employee Management | Add, edit, delete, and view employee details such as name, position, contact, etc. |
| Leave Application | Employees can apply for leave through a form-based interface. |
| Leave Approval | Admin can view, approve, or reject leave requests. |
| Dashboard | Overview of system stats like total employees, etc. |
| Logout | Secure session termination for each user. |

Non-Functional Requirements:

| Requirement Type | Specification |
|------------------|---|
| Usability | The system should have a user-friendly and responsive interface. |
| Scalability | The system should allow for future addition of new HR features. |
| Performance | System should respond to user actions within 2 seconds. |
| Security | User data and login credentials must be protected through proper authentication and encryption. |
| Availability | The system should be accessible 24/7 over the web |

Hardware Requirements:

| Component | Specification |
|-----------|------------------------------------|
| Processor | Intel i3 or higher |
| RAM | Minimum 4 GB. |
| Storage | Minimum 10 GB free disk space. |
| Display | 1024x768 resolution or higher |
| Internet | Required for deployment and access |

Software Requirements:

| Software | Specification |
|------------------|--|
| Operating System | Windows / Linux / macOS |
| Code Editor | Visual Studio Code. |
| API Testing Tool | Postman |
| Browser | Chrome, Firefox, or any modern browser |

Software model used

For the development of the **HR Management System**, the **Spiral Model** was chosen. This model combines the features of both the **Waterfall and Prototyping models**, making it ideal for medium to high-risk projects where frequent refinement and iterative development are required.

Spiral Model Overview:

The **Spiral Model** is a risk-driven software development process model that emphasizes repeated refinement through **iterations** or **phases**, called **spirals**. Each spiral includes the following activities:

1. **Planning**
2. **Risk Analysis**
3. **Engineering (Design & Development)**
4. **Evaluation**

Cost Estimation

The estimated cost for the HR Management System project is approximately ₹25,000. This includes the overall expenses for planning, design, development, testing, deployment, documentation, and maintenance. The project uses open-source technologies, which helps in minimizing licensing costs. The cost is primarily based on the time, resources, and effort required throughout the development lifecycle.

Project scheduling

| Phase | Duration |
|----------------------------------|----------|
| Requirement Gathering | 5 days |
| Design and Architecture | 5 days |
| Development (Frontend & Backend) | 1 month |
| Testing and Debugging | 1 week |
| Maintenance and Updates | Ongoing |

The project is expected to be completed in 7-8 weeks, with a regular review at the end of each phase to ensure that requirements are met and that any necessary adjustments are made. The maintenance phase will continue for ongoing updates and bug fixes.

Chapter -2

System Design

Project Planning:

The project planning phase outlines the tasks, timeline, resources, and methodology that will be followed to ensure the successful completion of the **HR Management System**. The planning is essential for organizing the workflow, ensuring that all requirements are met, and minimizing risks during development. It follows the **Spiral Model**, where iterative refinement and risk management are key to the process.

The planning phase involves defining the system's requirements, the scope of work, and the necessary resources, including both human resources and tools. A clear roadmap is created, outlining all the key milestones and deadlines for each phase of the project. This ensures that the team stays on track and delivers the system within the defined timeline.

The project is divided into several stages: requirement gathering, design, development, testing, deployment, and maintenance. Each phase has a dedicated timeline, and the team's responsibilities are clearly defined for each task. The project's success relies heavily on regular progress reviews, which will occur at the end of each spiral phase. This will allow for prompt feedback and adjustments to be made if necessary.

Modules:

The HR Management System is divided into various modules to ensure the smooth management of employee and leave-related data. Each module is designed to perform specific functions, making the system modular, maintainable, and scalable. The following are the key modules in the system:

Employee Management Module:

This module handles all aspects related to employee information. It allows the HR team to add, update, delete, and view employee details, such as personal information, contact details, department, designation, and more.

Key Features:

- Add new employee records
- View and search employee information
- Update employee records
- Delete employee records

Leave Management Module:

This module allows employees to apply for leaves and HR personnel to approve or reject leave requests. It tracks leave balances, leaves taken, for leave utilization.

Key Features:

- Apply for leave
- View leave history and balances
- HR approval and rejection
- Track leave status (approved, pending, rejected)

User Authentication and Role Management Module:

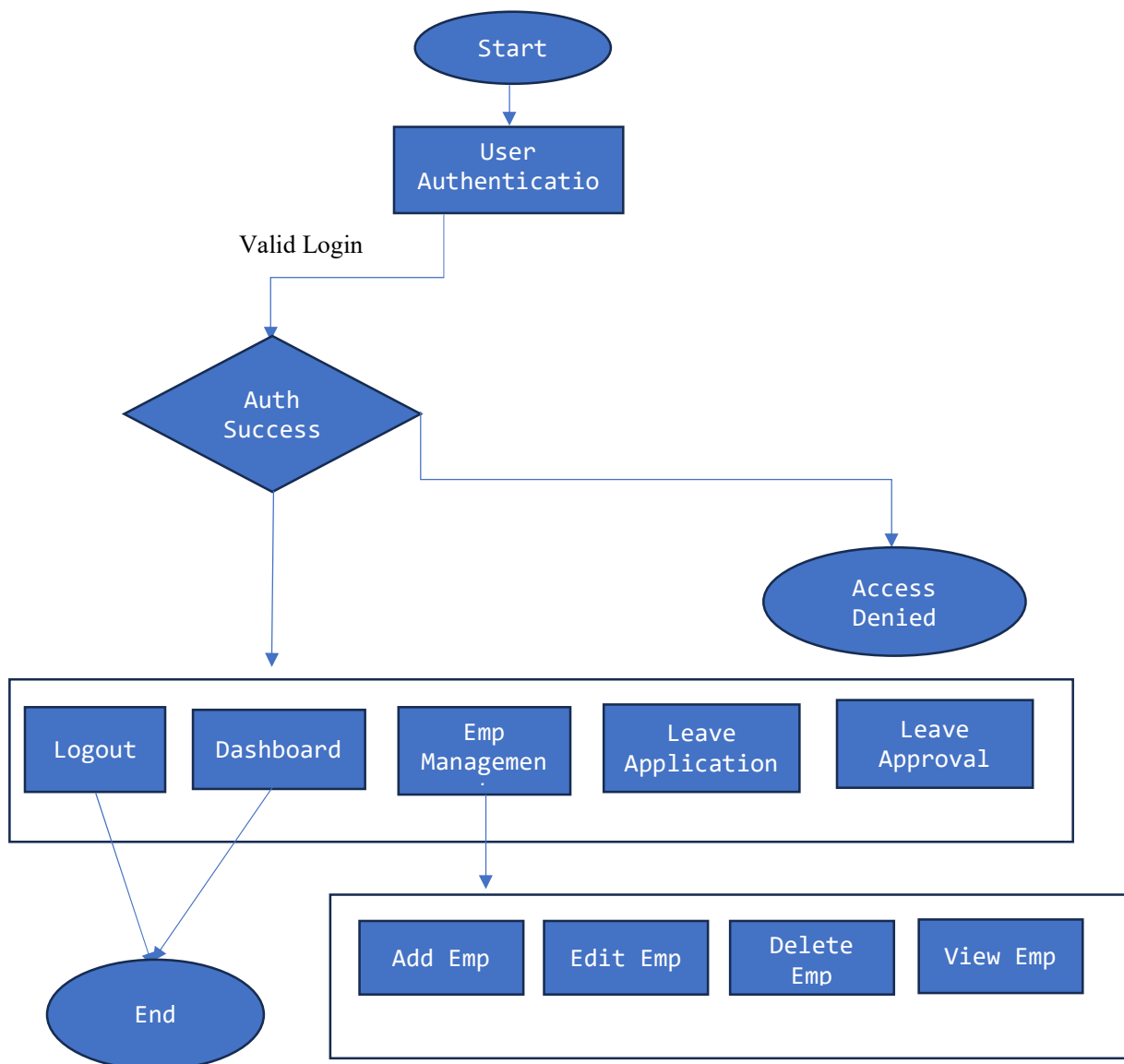
This module ensures secure access to the system by managing user roles (Admin, HR, Employee) and authenticating users.

Key Features:

- User login and registration
- Role-based access control
- Password management

Each module interacts with others to maintain a seamless flow of information and enhance user experience, all while ensuring secure and efficient data management.

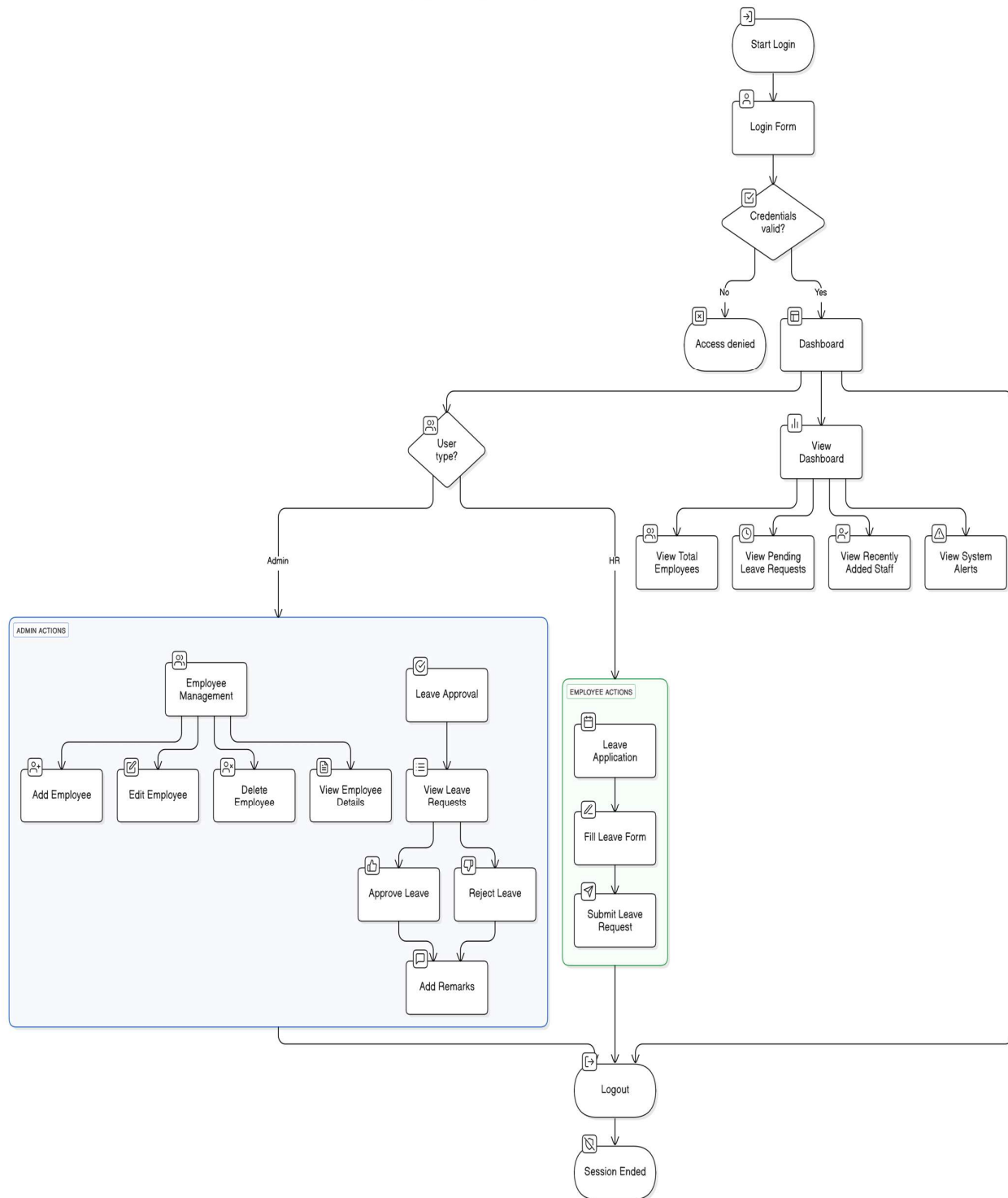
DFD (Data flow Diagram):



ER-diagram:

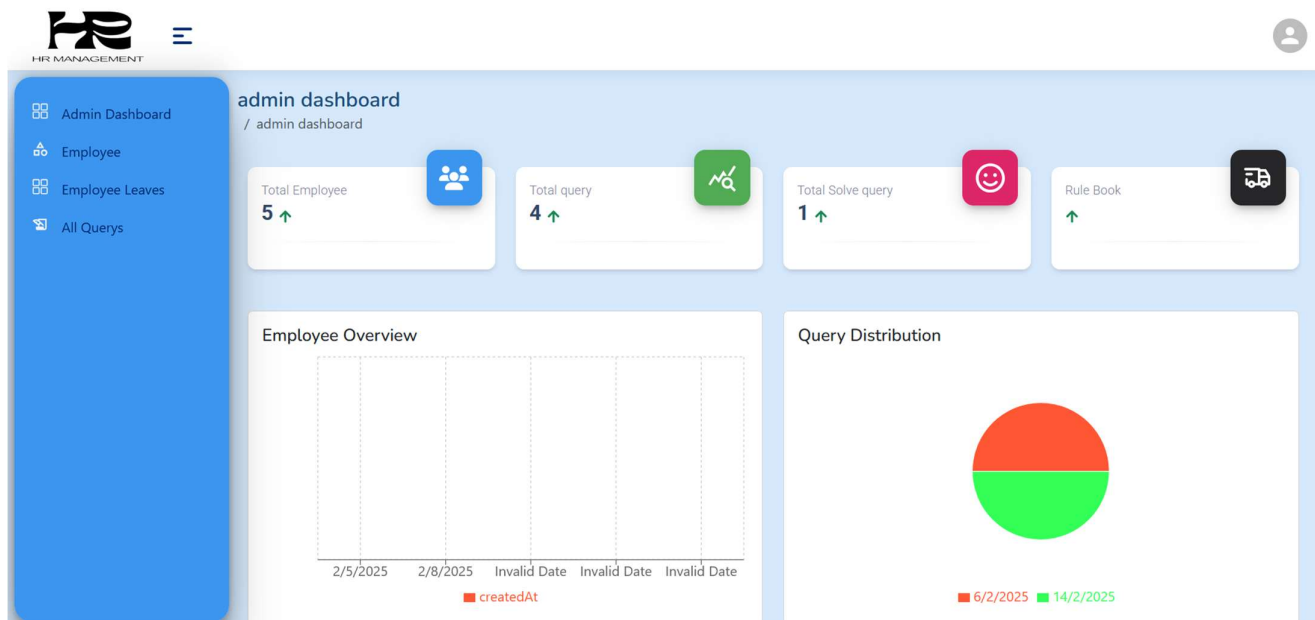
Figure 1

Employee Management System Flow Chart



Screen shots:

Dashboard:



Add Employee:

The screenshot shows the 'Add Employee' form in the HR Management system. The sidebar on the left has navigation links: Admin Dashboard, Employee, Add Employee, Employee list, Employee Leaves, and All Querys. The main content area has a header with the HR Management logo and a user profile icon. Below the header, there is a form titled 'Add Employee' with the following fields: Employee ID *, Name *, Employee Department *, Employee Salary *, Email *, Password *, Phone Number *, Address, City, State, Zip Code *, Joining Date * (dd / mm / yyyy), Casual Leave *, Sick Leave *, Upload Pan Card (Browse... No file selected.), Upload Resume (Browse... No file selected.), and Experience Letter (Browse... No file selected.). A tooltip 'Please fill out this field.' is visible over the City field. At the bottom right of the form are two buttons: CANCEL and SUBMIT.

Employee List:



 Admin Dashboard

 Employee

Add Employee

Employee list

 Employee Leaves

 All Querys




Employee List


/ Employee List


| <input type="checkbox"/> | Employee ID | Name | Email | Phone | Total CL | Total SL | Used CL |
|--------------------------|----------------|-----------------------|-----------------------------------|------------|----------|----------|---------|
| <input type="checkbox"/> | Dreamemp001 | Ankit Kumar Prajapati | ankitkumar@dreambytesolution.c... | 9288301902 | -2 | 0 | 2 |
| <input type="checkbox"/> | Dreamemp004 | Vikas | viki@gmail.com | 7282898388 | 5 | 332 | 18 |
| <input type="checkbox"/> | dreamadmin0012 | Ak kumar | ak@gmail.com | 7880966304 | 10 | 7 | 0 |
| <input type="checkbox"/> | 37 | mnjn | ankit@gmail.com | 7373834843 | 78 | 73 | 0 |
| <input type="checkbox"/> | 87 | ankit | abc@gmail.com | 7877887888 | 21 | 5 | 0 |

Rows per page: 100 1-5 of 5 < >

Update Employee:





 Admin Dashboard

 Employee

Add Employee

Employee list

 Employee Leaves

 All Querys

Employee List

/ Employee List

| State | Total CL |
|---------------|----------|
| uttar Pradesh | 9 |
| jahjh | 7 |
| Uttra Khand | 8 |
| jhs | 9 |
| jhhjk | 8 |

1 row selected

Update Employee Details

Ankit Kumar Prajapati

9288301902

Kanpur

KNN

uttar Pradesh

929999

Total CL
-2

Total SL

No file selected.











No file selected.

No file selected.

No file selected.

CANCEL

UPDATE

| Resume | Experience Letter | Actions |
|-----------------------------|--------------------------------------|---|
| View Resume | View Experience L... |   |
| View Resume | View Experience L... |   |
| View Resume | View Experience L... |   |
| View Resume | View Experience L... |   |
| View Resume | View Experience L... |   |

Rows per page: 100 1-5 of 5 < >

Employee Leaves:



HR MANAGEMENT

Admin Dashboard

Employee

Add Employee

Employee list

Employee Leaves


All Querys

Employee Leaves

/ Employee Leaves

| <input type="checkbox"/> | ID | Employee ID | Name | Email | Phone | Leave Type | Start Date | End Da |
|--------------------------|---------------------|-------------|-----------------------|-------------------------------|------------|------------|------------|--------|
| <input type="checkbox"/> | 67a730f063d59ce... | Dreamemp001 | Ankit Kumar Prajapati | ankitkumar@dreambytesoluti... | 9288301902 | CL | 10/2/2025 | 11/2/2 |
| <input type="checkbox"/> | 67a73c956c21ba8... | Dreamemp004 | Vikas | viki@gmail.com | 7282898388 | CL | 10/2/2025 | 13/2/2 |
| <input type="checkbox"/> | 67a73cf46c21ba8... | Dreamemp004 | Vikas | viki@gmail.com | 7282898388 | CL | 11/2/2025 | 21/2/2 |
| <input type="checkbox"/> | 67a741d776df4c8... | Dreamemp004 | Vikas | viki@gmail.com | 7282898388 | CL | 9/2/2025 | 11/2/2 |
| <input type="checkbox"/> | 67a7431176df4c8... | Dreamemp005 | ashish | ashish@gmail.com | 7823783783 | CL | 9/2/2025 | 12/2/2 |
| <input type="checkbox"/> | 67a744b976df4c8... | Dreamemp005 | ashish | ashish@gmail.com | 7823783783 | SL | 9/2/2025 | 13/2/2 |
| <input type="checkbox"/> | 67a746c776df4c8... | Dreamemp005 | ashish | ashish@gmail.com | 7823783783 | CL | 9/2/2025 | 11/2/2 |
| <input type="checkbox"/> | 67a09b15ff76dc2a... | Dreamemp005 | ashish | ashish@gmail.com | 7823783783 | CL | 10/2/2025 | 11/2/2 |

Rows per page: 100 1-12 of 12 < >



HR MANAGEMENT

Admin Dashboard

Employee

Add Employee

Employee list

Employee Leaves

All Querys




Employee Leaves


/ Employee Leaves


| | Phone | Leave Type | Start Date | End Date | Total Days | Status | Actions |
|-----------------|------------|------------|------------|-----------|------------|----------|----------------------------------|
| ambytesoluti... | 9288301902 | CL | 10/2/2025 | 11/2/2025 | 2 | Approved | <div>APPROVE REJECT UPDATE</div> |
| | 7282898388 | CL | 10/2/2025 | 13/2/2025 | 4 | Approved | <div>APPROVE REJECT UPDATE</div> |
| | 7282898388 | CL | 11/2/2025 | 21/2/2025 | 11 | Approved | <div>APPROVE REJECT UPDATE</div> |
| | 7282898388 | CL | 9/2/2025 | 11/2/2025 | 3 | Approved | <div>APPROVE REJECT UPDATE</div> |
| om | 7823783783 | CL | 9/2/2025 | 12/2/2025 | 4 | Rejected | <div>APPROVE REJECT UPDATE</div> |
| om | 7823783783 | SL | 9/2/2025 | 13/2/2025 | 5 | Rejected | <div>APPROVE REJECT UPDATE</div> |
| om | 7823783783 | CL | 9/2/2025 | 11/2/2025 | 3 | Rejected | <div>APPROVE REJECT UPDATE</div> |
| om | 7823783783 | CL | 10/2/2025 | 11/2/2025 | 2 | Rejected | <div>APPROVE REJECT UPDATE</div> |

Rows per page: 100 1-12 of 12 < >

Employee Queries:





 Admin Dashboard

 Employee

Add Employee

Employee list

 Employee Leaves

 All Querys

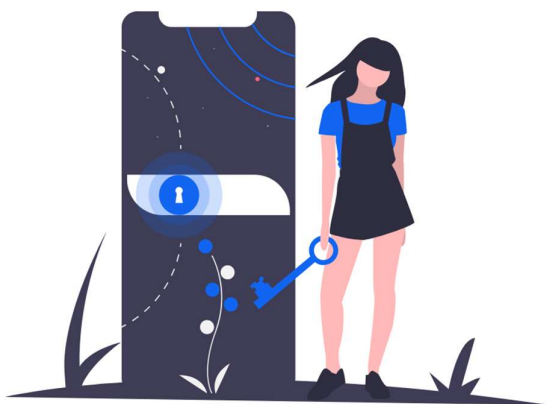
Employee Queries

/ Employee Queries

| <input type="checkbox"/> | S.No | Query ID | Query Subject | Query Message | Employee Name | Query Status |
|--------------------------|------|---------------------------|-----------------------------|---|-----------------------|--------------|
| <input type="checkbox"/> | 1 | 67a45d50c1a027395b936... | This is me ankit | this is my query | Ankit Kumar Prajapati | solve |
| <input type="checkbox"/> | 2 | 67a4aa048f42a2fd5d6db2... | This is my new query | Please solve this | Ankit Kumar Prajapati | unsolve |
| <input type="checkbox"/> | 3 | 67af28ab119ad20d002f4f... | this is my leave query test | please check | ankit | unsolve |
| <input type="checkbox"/> | 4 | 67af297d119ad20d002f4f... | this is test new | Lorem Ipsum is simply dummy text of the ... | ankit | unsolve |

Rows per page: 100 1~4 of 4

Login Page:



Coding:

Admin Model:

```
const mongoose = require('mongoose');

const AdminSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true,
    unique: true
  },
  phone: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true,
    minlength: 8,
  },
  adminId: {
    type: String
  },
  role: {
    type: String,
    default: "admin"
  },
  createdAt: {
    type: Date,
    default: Date.now
  }
});

module.exports = mongoose.model('Admin', AdminSchema);
```

Employee Model:

```
// const mongoose = require("mongoose");

// const EmployeeSchema = mongoose.Schema({
//   employeeId: {
//     type: String,
//     required: true,
//     unique: true,
//   },
//   name: {
//     type: String,
//     required: true,
//   },
//   email: {
//     type: String,
//     required: true,
//   },
//   password: {
//     type: String,
//     required: true,
//     minlength: 8,
//   },
//   phone: {
//     type: String,
//     required: true,
//     minlength: 10,
//   },
//   address1: {
//     type: String,
//     required: true,
//   },
//   state: {
//     type: String,
//     required: true,
//   },
//   city: {
//     type: String,
//     required: true,
//   },
//   zipcode: {
//     type: String,
//     required: true,
//   },
//   createdAt: {
//     type: Date,
//     default: Date.now,
//   },
//   role: {
//     type: String,
//     default: "employee",
//   },
//   department: {
//     type: String,
```

```

// },
// salary: {
//   type: Number,

// },
// CI: {
//   type: Number,
//   required: true,
//   default: 0
// },
// SI: {
//   type: Number,
//   required: true,
//   default: 0
// },
// joindate: {
//   type: Date,

// },
// is_deleted: {
//   type: Boolean,
//   default: false,
// },
// });

// module.exports = mongoose.model("Employee", EmployeeSchema);

const mongoose = require("mongoose");

const EmployeeSchema = mongoose.Schema({
  employeeId: {
    type: String,
    required: true,
    unique: true,
  },
  name: {
    type: String,
    required: true,
  },
  email: {
    type: String,
    required: true,
  },
  password: {
    type: String,
    required: true,
    minlength: 8,
  },
  phone: {
    type: String,
    required: true,
    minlength: 10,
  },
  address1: {
    type: String,

```

```
    required: true,
  },
  state: {
    type: String,
    required: true,
  },
  city: {
    type: String,
    required: true,
  },
  zipcode: {
    type: String,
    required: true,
  },
  department: {
    type: String,
  },
  salary: {
    type: Number,
  },
  Cl: {
    type: Number,
    required: true,
    default: 0
  },
  Sl: {
    type: Number,
    required: true,
    default: 0
  },
  joindate: {
    type: Date,
  },
  is_deleted: {
    type: Boolean,
    default: false,
  },
  createdAt: {
    type: Date,
    default: Date.now,
  },
  role: {
    type: String,
    default: "employee",
  },
  adhar: {
    type: String,
  },
  pan: {
    type: String,
  },
  resume: {
    type: String,
  },
  experience_letter: {
```

```
    type: String,  
  },  
});
```

```
module.exports = mongoose.model("Employee", EmployeeSchema);
```

Leave Model:

```
const mongoose = require("mongoose");

const LeaveSchema = mongoose.Schema({
  employeeId: { type: mongoose.Schema.Types.ObjectId, ref: "Employee", required: true },
  leaveType: { type: String, enum: ["CL", "SL"], required: true },
  startDate: { type: Date, required: true },
  endDate: { type: Date, required: true },
  totalDays: { type: Number, required: true },
  status: { type: String, enum: ["Pending", "Approved", "Rejected"], default: "Pending" },
  appliedAt: { type: Date, default: Date.now },
});

module.exports=new mongoose.model("Leave",LeaveSchema);
```

Query Model:

```
const mongoose = require("mongoose");

const QuerySchema = mongoose.Schema({
  subject: {
    type: String,
    required: true,
  },
  message: {
    type: String,
    required: true,
  },
  employee: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Employee',
    required: true,
  },
  status: {
    type: String,
    enum: ["solve", "unsolve"],
    default: "unsolve"
  },
  createdAt: {
    type: Date,
    default: Date.now,
  }
});

module.exports = mongoose.model("Query", QuerySchema);
```


Admin Controller (where all function or controller made which work done my admin)

```
const Admin = require("../models/admin");
const Employee = require("../models/employee");
const jwt = require("jsonwebtoken");
const bcrypt = require("bcryptjs");
const Query = require("../models/query");
const Leave = require("../models/leave");
const multer = require("multer");
const path = require("path");
const RegisterAdmin = async (req, res) => {
  const { name, email, phone, password, adminId } = req.body;

  try {
    if (!email || !phone || !password) {
      return res.status(400).json({
        error: true,
        message: "Something went wrong || Missing field",
      });
    }
  }

  // Hash the password

  const newAdmin = new Admin({
    name,
    email,
    phone,
    password,
    adminId,
  });

  await newAdmin.save();
  // console.log("Register Admin", newAdmin);

  return res.status(200).json({
    error: false,
    message: "Admin Registered successfully.",
    data: [newAdmin],
  });
} catch (error) {
  // console.log(error);
  return res.status(500).json({
    error: true,
    message: "Internal server error",
  });
}
};

// register employee

// const EmployeeRegister = async (req, res) => {
//   const {
//     employeeId,
```

```
// name,
// email,
// phone,
// password,
// address1,
// state,
// city,
// zipcode,
// department,
// Cl,
// Sl,
// salary,
// joindate,
// } = req.body;

// try {
//   if (
//     !email ||
//     !phone ||
//     !name ||
//     !password ||
//     !employeeId ||
//     !Cl ||
//     !Sl ||
//     !salary ||
//     !department ||
//     !joindate
//   ) {
//     return res.status(400).json({
//       error: true,
//       message: "Required fields are missing",
//     });
//   }

//   const newemployee = new Employee({
//     employeeId,
//     name,
//     email,
//     phone,
//     password,
//     address1,
//     state,
//     city,
//     zipcode,
//     department,
//     Cl,
//     Sl,
//     salary,
//     joindate,
//   });

//   await newemployee.save();

//   return res.status(201).json({
//     error: false,
```

```
// message: "Employee registered successfully",
// data: newemployee, // Return the entire saved document
// });
// } catch (error) {
// console.error(error);
// return res.status(500).json({
// error: true,
// message: "Internal server error",
// });
// }
// };
```

Employee Register:

```
const EmployeeRegister = async (req, res) => {
  try {
    const {
      employeeId,
      name,
      email,
      phone,
      password,
      address1,
      state,
      city,
      zipcode,
      department,
      CI,
      SI,
      salary,
      joindate,
    } = req.body;

    if (
      !email ||
      !phone ||
      !name ||
      !password ||
      !employeeId ||
      !CI ||
      !SI ||
      !salary ||
      !department ||
      !joindate
    ) {
      return res.status(400).json({
        error: true,
        message: "Required fields are missing",
      });
    }
  }
}
```

```

// Get uploaded file paths (if present)
const adhar = req.files["adhar"] ? req.files["adhar"][0].path : null;
const pan = req.files["pan"] ? req.files["pan"][0].path : null;
const resume = req.files["resume"] ? req.files["resume"][0].path : null;
const experience_letter = req.files["experience_letter"]
  ? req.files["experience_letter"][0].path
  : null;

const newEmployee = new Employee({
  employeeId,
  name,
  email,
  phone,
  password,
  address1,
  state,
  city,
  zipcode,
  department,
  Cl,
  Sl,
  salary,
  joindate,
  adhar,
  pan,
  resume,
  experience_letter,
});

await newEmployee.save();

return res.status(201).json({
  error: false,
  message: "Employee registered successfully",
  data: newEmployee,
});
} catch (error) {
  console.error(error);
  return res.status(500).json({
    error: true,
    message: "Internal server error",
  });
}
};

```

User and admin Login:

```
const loginUser = async (req, res) => {
  const { adminId, employeeId, password } = req.body;

  try {
    let user;
    let role;

    // Check if the user is logging in using adminId or employeeId
    if (adminId) {

      user = await Admin.findOne({ adminId });
      role = "admin";
    } else if (employeeId) {

      user = await Employee.findOne({ employeeId });
      role = "employee";
    } else {
      return res.status(400).json({
        error: true,
        message: "Please provide either adminId or employeeId",
      });
    }

    // If no user is found
    if (!user) {
      return res.status(404).json({
        error: true,
        message: "User not found",
      });
    }

    // Compare password (plaintext comparison for now)
    if (password !== user.password) {
      return res.status(401).json({
        error: true,
        message: "Invalid credentials",
      });
    }

    // Generate JWT token
    const token = jwt.sign(
      { id: user._id, role },
      process.env.JWT_SECRET,
      { expiresIn: "1h" }
    );

    // Send response with user details and token
    return res.status(200).json({
      error: false,
      message: "Login successful",
      name: user.name,
    });
  }
}
```

```

    role,
    token,
  });
} catch (error) {
  // console.log(error);
  return res.status(500).json({
    error: true,
    message: "Internal server error",
  });
}
};

```

Get all employee leaves:

```

const Getemployeeswithleaves = async (req, res) => {
  try {
    const employeeelist = await Employee.find({ is_deleted: { $ne: 1 } });

    if (!employeeelist || employeeelist.length === 0) {
      return res.status(404).json({
        error: true,
        message: "No Employee found",
      });
    }

    // Calculate remaining CL & SL leaves for each employee
    const employeesWithLeaves = await Promise.all(
      employeeelist.map(async (employee) => {
        // Fetch the most recent leave details from the Employee collection
        const latestEmployee = await Employee.findById(employee._id);

        // Get total CL and SL leaves taken from Leave collection
        const leaves = await Leave.find({
          employeeId: employee._id,
        });

        let clUsed = 0;
        let slUsed = 0;
        let clApplied = 0;
        let slApplied = 0;

        // Count how many leaves are used for each type
        leaves.forEach((leave) => {
          if (leave.status === "Approved") {
            if (leave.leaveType === "CL") {
              clUsed += leave.totalDays;
            } else if (leave.leaveType === "SL") {
              slUsed += leave.totalDays;
            }
          }
        });
      })
    );
  }
};

```

```

    }
    } else if (leave.status === "Pending") {
    if (leave.leaveType === "CL") {
        clApplied += leave.totalDays;
    } else if (leave.leaveType === "SL") {
        slApplied += leave.totalDays;
    }
    }
    }
    });

// Ensure CL and SL values are always correct and dynamic
const totalCL = latestEmployee.Cl || 0;
const totalSL = latestEmployee.Sl || 0;
const remainingCL = Math.max(totalCL - clUsed, 0);
const remainingSL = Math.max(totalSL - slUsed, 0);

return {
    _id: latestEmployee._id,
    employeeId: latestEmployee.employeeId,
    name: latestEmployee.name,
    email: latestEmployee.email,
    phone: latestEmployee.phone,
    address1: latestEmployee.address1,
    city: latestEmployee.city,
    state: latestEmployee.state,
    zipcode: latestEmployee.zipcode,
    role: latestEmployee.role,
    createdAt: latestEmployee.createdAt.toLocaleString(),
    department: latestEmployee.department,
    adhar: latestEmployee.adhar,
    pan: latestEmployee.pan,
    resume: latestEmployee.resume,
    experience_letter: latestEmployee.experience_letter,
    salary: latestEmployee.salary,
    Cl: totalCL,
    Sl: totalSL,
    usedCL: clUsed,
    usedSL: slUsed,
    appliedCL: clApplied,
    appliedSL: slApplied,
    remainingCL,
    remainingSL,
};
})
);

return res.status(200).json({
    error: false,
    message: "Employee list with leave details",
    data: employeesWithLeaves,
});
} catch (error) {
    console.error("Error fetching employee leaves:", error);
    return res.status(500).json({
        error: true,

```

```
    message: "Internal server error",
  });
}
};
```

// get all leaves

```
const getAllLeaveRequests = async (req, res) => {
  try {
    const leaveRequests = await Leave.find().populate(
      "employeeId",
      "employeeId name email phone department salary CI SI"
    );

    res.status(200).json({ error: false, data: leaveRequests });
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: true, message: "Internal Server Error" });
  }
};
```

// // Update employee

```
const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    cb(null, "uploads/"); // Upload files to "uploads" directory
  },
  filename: (req, file, cb) => {
    cb(null, `${Date.now()}-${file.originalname}`); // Unique file names
  },
});

// File filter to allow only specific file types
const fileFilter = (req, file, cb) => {
  const allowedTypes = ["image/jpeg", "image/png", "application/pdf"];
  if (allowedTypes.includes(file.mimetype)) {
    cb(null, true);
  } else {
    cb(
      new Error("Invalid file type. Only JPG, PNG, and PDF are allowed."),
      false
    );
  }
};

// Multer middleware
const upload = multer({
  storage: storage,
  fileFilter: fileFilter,
});
```


Update employee:

```
const Updateemployee = async (req, res) => {
  try {
    const { id } = req.params;
    const {
      name,
      phone,
      address1,
      city,
      state,
      zipcode,
      department,
      salary,
      remainingCL,
      remainingSL,
      Cl,
      Sl,
      usedCL,
      usedSL,
    } = req.body;

    const updateData = {
      name,
      phone,
      address1,
      city,
      state,
      zipcode,
      department,
      salary,
      remainingCL,
      remainingSL,
      Cl,
      Sl,
      usedCL,
      usedSL,
    };

    // Handling optional file uploads
    if (req.files) {
      if (req.files.adhar) updateData.adhar = req.files.adhar[0].path;
      if (req.files.pan) updateData.pan = req.files.pan[0].path;
      if (req.files.resume) updateData.resume = req.files.resume[0].path;
      if (req.files.experience_latter)
        updateData.experience_latter = req.files.experience_latter[0].path;
    }

    // Update the employee record
    const updatedEmployee = await Employee.findByIdAndUpdate(id, updateData, {
      new: true,
      runValidators: true,
    });
  }
};
```

```

if (!updatedEmployee) {
  return res.status(404).json({
    error: true,
    message: "Employee not found or not updated.",
  });
}

return res.status(200).json({
  error: false,
  message: "Employee Updated Successfully.",
  data: updatedEmployee,
});
} catch (error) {
  return res.status(500).json({
    error: true,
    message: "Internal server error",
    details: error.message,
  });
}
};

```

Delete Employee:

```

const Delemployee = async (req, res) => {
  const { id } = req.params;

  try {
    if (!id) {
      return res.status(400).json({
        error: true,
        message: "Something went wrong || Employee Id missing",
      });
    }
    const delemp = await Employee.findByIdAndUpdate(
      id,
      { is_deleted: 1 },
      { new: true }
    );
    if (!delemp) {
      return res.status(400).json({
        error: true,
        message: "Employee not delete",
      });
    }
    return res.status(200).json({
      error: false,
      message: "Employee deleted successfully",
      data: delemp,
    });
  } catch (error) {
    return res.status(500).json({
      error: true,

```

```
    message: "Internal server error",
  });
}
};
```

// getallemployee Query

```
const Getallquerys = async (req, res) => {
  try {
    const querylist = await Query.find().populate("employee"); // Populate employee details

    if (!querylist || querylist.length === 0) {
      return res.status(404).json({
        error: true,
        message: "No queries found",
      });
    }

    return res.status(200).json({
      error: false,
      message: "Query list retrieved successfully",
      data: querylist,
    });
  } catch (error) {
    console.error("Error fetching queries:", error);
    return res.status(500).json({
      error: true,
      message: "Internal server error",
    });
  }
};
```

Get all solve query:

```
const Getallsolvequerys = async (req, res) => {
  try {
    const querylist = await Query.find({ status: "solve" });
    if (!querylist || querylist.length === 0) {
      return res.status(404).json({
        error: true,
        message: "No query here",
      });
    }
    return res.status(200).json({
      error: false,
      message: "Query list",
      data: querylist,
    });
  } catch (error) {
```

```
return res.status(500).json({
  error: true,
  message: "Internal server error",
});
}
```

// update query status:

```
const Updatequerystatus = async (req, res) => {
  const { id } = req.params;
  const { status } = req.body;
  try {
    if (!id) {
      return res.status(400).json({
        error: true,
        message: "Something went wrong || ID missing ",
      });
    }
    const data = await Query.findByIdAndUpdate(
      id,
      { status: "solve" },
      { new: true }
    );

    if (!data) {
      return res.status(404).json({
        error: true,
        message: "Data not Updated",
      });
    }
    return res.status(200).json({
      error: true,
      message: "Query status Updated",
      data: data,
    });
  } catch (error) {
    return res.status(500).json({
      error: true,
      message: "Internal server error",
    });
  }
};
```

Logout User:

```
const logoutUser = async (req, res) => {
  try {
    return res.status(200).json({
      error: false,
      message: "User logged out successfully",
    });
  } catch (error) {
    console.error(error);
    return res.status(500).json({
      error: true,
      message: "Internal server error",
    });
  }
};
```

Leave Approve:

```
const approveLeave = async (req, res) => {
  const { leaveId, status } = req.body;

  try {
    // Find the leave request
    const leave = await Leave.findById(leaveId);
    if (!leave) {
      return res
        .status(404)
        .json({ error: true, message: "Leave request not found" });
    }

    // Validate status
    if (status !== "Approved" && status !== "Rejected") {
      return res.status(400).json({ error: true, message: "Invalid status" });
    }

    // Update leave status
    leave.status = status;
    await leave.save();

    // If status is "Approved", update employee's leave balance
    if (status === "Approved") {
      const employee = await Employee.findById(leave.employeeId);
      if (!employee) {
        return res
          .status(404)
          .json({ error: true, message: "Employee not found" });
      }
    }
  }
};
```

```

// Ensure employee.Cl and employee.Sl are valid numbers
if (leave.leaveType === "CL" && typeof employee.Cl === "number") {
  employee.Cl -= leave.totalDays;
} else if (leave.leaveType === "SL" && typeof employee.Sl === "number") {
  employee.Sl -= leave.totalDays;
}

// Save the updated employee
await employee.save();
}

return res
  .status(200)
  .json({ error: false, message: `Leave request ${status}`, data: leave });
} catch (error) {
  console.error(error);
  return res
    .status(500)
    .json({ error: true, message: "Internal server error" });
}
};

```

// update applied leave

```

const updateLeave = async (req, res) => {
  const { leaveId } = req.params;
  const { employeeId, leaveType, startDate, endDate } = req.body;

  try {
    const leave = await Leave.findById(leaveId);
    if (!leave) {
      return res
        .status(404)
        .json({ error: true, message: "Leave request not found" });
    }

    if (leave.status !== "Pending") {
      return res
        .status(400)
        .json({
          error: true,
          message: "Only pending leave requests can be updated",
        });
    }

    const employee = await Employee.findById(employeeId);
    if (!employee) {
      return res
        .status(404)
        .json({ error: true, message: "Employee not found" });
    }
  }
};

```

```

const totalDays =
  Math.ceil(
    (new Date(endDate) - new Date(startDate)) / (1000 * 60 * 60 * 24)
  ) + 1;

if (leaveType === "CL" && employee.Cl < totalDays) {
  return res
    .status(400)
    .json({ error: true, message: "Insufficient Casual Leave balance" });
}
if (leaveType === "SL" && employee.Sl < totalDays) {
  return res
    .status(400)
    .json({ error: true, message: "Insufficient Sick Leave balance" });
}

leave.leaveType = leaveType;
leave.startDate = startDate;
leave.endDate = endDate;
leave.totalDays = totalDays;

await leave.save();

return res
  .status(200)
  .json({
    error: false,
    message: "Leave request updated successfully",
    data: leave,
  });
} catch (error) {
  console.error(error);
  return res
    .status(500)
    .json({ error: true, message: "Internal server error" });
}
};

module.exports = {
  RegisterAdmin,
  loginUser,
  Employeeregister,
  Getemployeeswithleaves,
  Updateemployee,
  Delemployee,
  Getallquerys,
  Getallsolvequerys,
  Updatequerystatus,
  logoutUser,
  getAllLeaveRequests,
  approveLeave,
  updateLeave,
};

```

Employee Controller:

```
const Query = require("../models/query");
const Employee = require("../models/employee");
const Leave=require("../models/leave");
```

Add Query:

```
const Addquery = async (req, res) => {
  const { subject, message, employeeId } = req.body;

  try {

    if (!subject || !message || !employeeId) {
      return res.status(400).json({
        error: true,
        message: "Missing subject, message, or employee ID."
      });
    }

    const employee = await Employee.findById(employeeId);
    if (!employee) {
      return res.status(404).json({
        error: true,
        message: "Employee not found."
      });
    }

    const newQuery = new Query({
      subject,
      message,
      employee: employee._id
    });

    await newQuery.save();

    return res.status(200).json({
      error: false,
      message: "Your query was sent successfully.",
      data: newQuery
    });

  } catch (error) {
    return res.status(500).json({
      error: true,
      message: "Internal server error",
      details: error.message
    });
  }
};
```


Get all Query:

```
const GetAllQuery = async (req, res) => {
  const { employeeId } = req.params; // Get employeeId from URL params

  try {
    if (!employeeId) {
      return res.status(400).json({
        error: true,
        message: "Employee ID is required."
      });
    }

    // Find all queries where `employee` field matches the given employee ID
    const queries = await Query.find({ employee: employeeId }).populate("employee", "name email phone");

    if (!queries || queries.length === 0) {
      return res.status(404).json({
        error: true,
        message: "No queries found for this employee."
      });
    }

    return res.status(200).json({
      error: false,
      message: "Query list retrieved successfully.",
      data: queries
    });
  } catch (error) {
    return res.status(500).json({
      error: true,
      message: "Internal server error",
      details: error.message
    });
  }
};
```

// leave apply

```
const applyLeave = async (req, res) => {
  const { employeeId, leaveType, startDate, endDate } = req.body;

  try {
    const employee = await Employee.findById(employeeId);
    if (!employee) {
      return res.status(404).json({ error: true, message: "Employee not found" });
    }
  }
```

```

const totalDays = Math.ceil((new Date(endDate) - new Date(startDate)) / (1000 * 60 * 60 * 24)) + 1;

if (leaveType === "CL" && employee.Cl < totalDays) {
  return res.status(400).json({ error: true, message: "Insufficient Casual Leave balance" });
}
if (leaveType === "SL" && employee.Sl < totalDays) {
  return res.status(400).json({ error: true, message: "Insufficient Sick Leave balance" });
}

const newLeave = new Leave({
  employeeId,
  leaveType,
  startDate,
  endDate,
  totalDays,
  status: "Pending",
});

await newLeave.save();
return res.status(201).json({ error: false, message: "Leave request submitted successfully", data:
newLeave });
} catch (error) {
  console.error(error);
  return res.status(500).json({ error: true, message: "Internal server error" });
}
};

// leave history

const getLeaveHistory = async (req, res) => {
  const { employeeId } = req.params;

  try {
    const leaves = await Leave.find({ employeeId }).sort({ appliedAt: -1 });
    return res.status(200).json({ error: false, data: leaves });
  } catch (error) {
    console.error(error);
    return res.status(500).json({ error: true, message: "Internal server error" });
  }
};

module.exports = { Addquery, GetAllQuery, applyLeave, getLeaveHistory, };

```

Auth Middleware:

```
const jwt = require("jsonwebtoken");

const verifyToken = (req, res, next) => {
  const token = req.header("Authorization");

  if (!token) {
    return res.status(401).json({
      error: true,
      message: "Access denied. No token provided.",
    });
  }

  try {
    const decoded = jwt.verify(
      token.replace("Bearer ", ""),
      process.env.JWT_SECRET
    );
    req.user = decoded;
    next();
  } catch (error) {
    return res.status(401).json({
      error: true,
      message: "Invalid token.",
    });
  }
};

// Middleware for Admin Authorization
const isAdmin = (req, res, next) => {
  if (req.user.role !== "admin") {
    return res.status(403).json({
      error: true,
      message: "Access denied. Admins only.",
    });
  }
  next();
};

// Middleware for Employee Authorization
const isEmployee = (req, res, next) => {
  if (req.user.role !== "employee") {
    return res.status(403).json({
      error: true,
      message: "Access denied. Employees only.",
    });
  }
  next();
};

module.exports = { verifyToken, isAdmin, isEmployee };
```

Multer Configuration:

```
const multer = require("multer");
const path = require("path");

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "uploads/");
  },
  filename: function (req, file, cb) {
    cb(null, Date.now() + path.extname(file.originalname));
  },
});

const fileFilter = (req, file, cb) => {
  const allowedTypes = ["image/jpeg", "image/png", "application/pdf"];
  if (allowedTypes.includes(file.mimetype)) {
    cb(null, true); // Accept the file
  } else {
    cb(new Error("Only JPG, PNG, and PDF files are allowed"), false);
  }
};

const upload = multer({
  storage,
  fileFilter,
  limits: { fileSize: 10 * 1024 * 1024 },
});

module.exports = upload;
```

Admin Routes:

```
const express = require("express");
const router = express.Router();
const upload = require("../middleware/multerConfig");

const {
  RegisterAdmin,
  loginUser,
  Employeeregister,
  Getemployeeswithleaves,
  Updateemployee,
  Delemployee,
  GetAllquerys,
  Getallsolvequerys,
  Updatequerystatus,
  logoutUser,
  getAllLeaveRequests,
  approveLeave,
  updateLeave,
} = require("../controllers/admincontroller");

const { verifyToken, isAdmin } = require("../middleware/authMiddleware");

// Admin Registration & Login
router.post("/registeradmin", RegisterAdmin);
router.post("/login", loginUser);

// Employee Registration with Multer for file uploads
router.post(
  "/registeremp",
  upload.fields([
    { name: "adhar", maxCount: 10 },
    { name: "pan", maxCount: 10 },
    { name: "resume", maxCount: 10 },
    { name: "experience_letter", maxCount: 10 },
  ]),
  Employeeregister
);

// Employee Operations
router.get("/emplist", Getemployeeswithleaves, verifyToken, isAdmin);
// router.put("/updateemp/:id", Updateemployee);
router.put(
  "/updateemp/:id",
  upload.fields([
    { name: "adhar", maxCount: 10 },
    { name: "pan", maxCount: 10 },
    { name: "resume", maxCount: 10 },
    { name: "experience_latter", maxCount: 10 },
  ]),
  Updateemployee
);

router.delete("/delemp/:id", Delemployee, verifyToken, isAdmin);
```

```
// Queries Management
router.get("/getallquery", GetAllquerys);
router.get("/solvequery", GetAllsolvequerys);
router.put("/chnagestatus/:id", Updatequerystatus);

// Leave Management
router.post("/logout", logoutUser);
router.get("/allleaves", getAllLeaveRequests);
router.put("/leaveapprove", approveLeave);
router.put("/updateleave/:leaveId", updateLeave, verifyToken, isAdmin);

module.exports = router;
```

Employee Routes:

```
const express = require("express");
```

```
const router = express.Router();
```

```
const {  
  Addquery,  
  GetAllQuery,  
  applyLeave,  
  getLeaveHistory  
} = require("../controllers/employeecontroller");
```

```
const {  
  verifyToken,  
  isEmployee,  
} = require("../middleware/authMiddleware");
```

```
router.post("/addquery",Addquery, verifyToken,isEmployee);
```

```
router.get("/getquery/:employeeId",GetAllQuery);
```

```
router.post("/applyleave",applyLeave,verifyToken,isEmployee);
```

```
router.get("/leavehistory/:employeeId",getLeaveHistory,verifyToken,isEmployee);
```

```
module.exports = router;
```

Package Json:

```
{
  "name": "backend",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "cors": "^2.8.5",
    "dotenv": "^16.4.7",
    "express": "^4.21.2",
    "json": "^11.0.0",
    "jsonwebtoken": "^9.0.2",
    "mongoose": "^8.9.7",
    "multer": "^1.4.5-lts.1",
    "socket.io": "^4.8.1"
  }
}
```


Package-Lock. Json

```
{
  "name": "backend",
  "version": "1.0.0",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "backend",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "bcryptjs": "^2.4.3",
        "cors": "^2.8.5",
        "dotenv": "^16.4.7",
        "express": "^4.21.2",
        "json": "^11.0.0",
        "jsonwebtoken": "^9.0.2",
        "mongoose": "^8.9.7",
        "multer": "^1.4.5-lts.1",
        "socket.io": "^4.8.1"
      }
    },
    "node_modules/@mongodb-js/saslprep": {
      "version": "1.1.9",
      "resolved": "https://registry.npmjs.org/@mongodb-js/saslprep/-/saslprep-1.1.9.tgz",
      "integrity": "sha512-  
tVkljjeEaAhCqTzajSdgbQ6gE6f3oneVwa3iXR6csiEwXXOFsiC6Uh9iAjAhXPtqa/XMDHWjjeNH/77m/Y  
q2dw==",
      "dependencies": {
        "sparse-bitfield": "^3.0.3"
      }
    },
    "node_modules/@socket.io/component-emitter": {
      "version": "3.1.2",
      "resolved": "https://registry.npmjs.org/@socket.io/component-emitter/-/component-emitter-3.1.2.tgz",
      "integrity": "sha512-  
9BCxFwvbGg/RsZK9tjXd8s4UcwR0MWefQ1XEKIQQVVvAGJyINdrqKMcTRyLoK8Rse1GjzLV9cwjWV  
1oIXRWEXVA==",
    },
    "node_modules/@types/cors": {
      "version": "2.8.17",
      "resolved": "https://registry.npmjs.org/@types/cors/-/cors-2.8.17.tgz",
      "integrity": "sha512-  
8CGDvrBj1zgo2qE+oS3pOCyYNqCPryMWY2bGfwA0dcfopWGgxs+78df0Rs3rc9THP4JkOhLsAa+15Vd  
pAqkcUA==",
      "dependencies": {
        "@types/node": "*"
      }
    },
    "node_modules/@types/node": {
      "version": "22.13.1",
      "resolved": "https://registry.npmjs.org/@types/node/-/node-22.13.1.tgz",
      "integrity": "sha512-
```

```

jK8uzQlrVXqEU91UxiK5J7pKHyzgnl1Qnl0QDHIgVGuolJhRb9EEI28Cj9b3rGR8B2lhFCtvIm5os8IFnO/1
Ew==",
  "dependencies": {
    "undici-types": "~6.20.0"
  },
},
"node_modules/@types/webidl-conversions": {
  "version": "7.0.3",
  "resolved": "https://registry.npmjs.org/@types/webidl-conversions/-/webidl-conversions-7.0.3.tgz",
  "integrity": "sha512-CiJvcRtIlgzadHCYXw7dqEnMNRjhGZIYK05Mj9OyktqV8uVT8fD2BFOB7S1uwBE3Kj2Z+4UyPmFw/I
xgw/LAlA==",
},
"node_modules/@types/whatwg-url": {
  "version": "11.0.5",
  "resolved": "https://registry.npmjs.org/@types/whatwg-url/-/whatwg-url-11.0.5.tgz",
  "integrity": "sha512-coYR071JRaHa+xEvvYqvnIHAvqaYrLPbsufM9BF63HkwI5Lgmy2QR8Q5K/IYDY05AK82wOvSOS0U
sLTpTG7uQ==",
  "dependencies": {
    "@types/webidl-conversions": "*"
  },
},
"node_modules/accepts": {
  "version": "1.3.8",
  "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",
  "integrity": "sha512-2Y7UxjDvBcLkxw89PN9knsI02U6qVIx6nE0gXqR+4630XD26k1uBzHvZuXZmPz3iG+Dn6qFh0c/PyUw==",
  "dependencies": {
    "mime-types": "~2.1.34",
    "negotiator": "0.6.3"
  },
  "engines": {
    "node": ">= 0.6"
  },
},
"node_modules/append-field": {
  "version": "1.0.0",
  "resolved": "https://registry.npmjs.org/append-field/-/append-field-1.0.0.tgz",
  "integrity": "sha512-klpgFSWLW1ZEs8svjfb7g4qWY0YS5imI82dTg+QahUvJ8YqAY0P10Uk8tTyh9ZGuYEZEMaeJYCF5BF
uX552hsw==",
},
"node_modules/array-flatten": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",
  "integrity": "sha512-PCVAQswWemu6UdxdDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nNIM++NqRcK6CxxpUaf
jmhIdKiHibqg==",
},
"node_modules/base64id": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/base64id/-/base64id-2.0.0.tgz",
  "integrity": "sha512-4PUwK8I6JmKDeZ8XGsfIaM+bKxNtoHAOG0N4Y0Vb418CDfUpqK4bBH6UM3XhhUmtt3ZLkKMPb

```

```
lGe34o6EHj9y3Kts9R4ZYs/Gr+6N7MCAmIIFA3F1R2O5/m7K06AxfSeO5530PEERE6/WyEg3lsuyw4GH
IPZHog==",
  "engines": {
    "node": "^4.5.0 || >= 5.9"
  }
},
"node_modules/bcryptjs": {
  "version": "2.4.3",
  "resolved": "https://registry.npmjs.org/bcryptjs/-/bcryptjs-2.4.3.tgz",
  "integrity": "sha512-V/Hy/X9Vt7f3BbPJEi8BdVFMBYHi+jNXrYkW3huaybV/kQ0KJg0Y6PkEMbn+zeT+i+SiKZ/HMqJGIIt4L
ZDqNQ==",
},
"node_modules/body-parser": {
  "version": "1.20.3",
  "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.20.3.tgz",
  "integrity": "sha512-W702H2U0Dd1K8Hx4G891XWp1Y8QO4OYSN4qYl8YU71Dh3D0h30wV3D8B8wWzpmX8Q8xg3U5r+D8fA
7rAxByjUMqQ3/bHJy7D6OGXvx/MMc4IqBn/X0fcM1QUcAItpZrBEYhWGEm+tzXH90c+G01ypMcYJB
O9Y30203g==",
  "dependencies": {
    "bytes": "3.1.2",
    "content-type": "~1.0.5",
    "debug": "2.6.9",
    "depd": "2.0.0",
    "destroy": "1.2.0",
    "http-errors": "2.0.0",
    "iconv-lite": "0.4.24",
    "on-finished": "2.4.1",
    "qs": "6.13.0",
    "raw-body": "2.5.2",
    "type-is": "~1.6.18",
    "unpipe": "1.0.0"
  },
  "engines": {
    "node": ">= 0.8",
    "npm": "1.2.8000 || >= 1.4.16"
  }
},
"node_modules/bson": {
  "version": "6.10.2",
  "resolved": "https://registry.npmjs.org/bson/-/bson-6.10.2.tgz",
  "integrity": "sha512-5afhLTjqDSA3akH56E+/2J6kTDuSIlBxyXPdQslj9hcIgOUE378xdOfZvC/9q3LifJNl6KR/juZ+d0NRNYB
wXg==",
  "engines": {
    "node": ">=16.20.1"
  }
},
"node_modules/buffer-equal-constant-time": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/buffer-equal-constant-time/-/buffer-equal-constant-time-
1.0.1.tgz",
  "integrity": "sha512-zRpUjDwd/xk6ADqPMATG8vc9VPrckk7T07OIx0gnjmJAnHnTVXNQG3vfvWNuiZlkwu9KrKdA1iJKfsf
TVxE6NA=="
}
```

```

},
"node_modules/buffer-from": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/buffer-from/-/buffer-from-1.1.2.tgz",
  "integrity": "sha512-E+XQCRwSbaaiChtv6k6Dwgc+bx+Bs6vuKJHHI5kox/BaKbhiXzqQOwK4cO22yElGp2OCmjlVhT3HmxgyPGnJfQ==",
},
"node_modules/busboy": {
  "version": "1.6.0",
  "resolved": "https://registry.npmjs.org/busboy/-/busboy-1.6.0.tgz",
  "integrity": "sha512-8SFQbg/0hQ9xy3UNTB0YEnsNBbWfhf7RtnzpL7TtKiTBRfrQ9Fxcnz7VJsleJpyp6rVLvXiuORqjlHi5q+PYuA==",
  "dependencies": {
    "streamsearch": "^1.1.0"
  },
  "engines": {
    "node": ">=10.16.0"
  }
},
"node_modules/bytes": {
  "version": "3.1.2",
  "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",
  "integrity": "sha512-/Nf7TyzTx6S3yRJOboOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBdcYbEg==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/call-bind-apply-helpers": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/call-bind-apply-helpers/-/call-bind-apply-helpers-1.0.1.tgz",
  "integrity": "sha512-BhYE+WDaywFg2TBWYNXAE+8B1ATnThNBqXHP5nQu0jWJdVvY2hvkpyB3qOmtmDePiS5/BDQ8wASEWGMWRG148g==",
  "dependencies": {
    "es-errors": "^1.3.0",
    "function-bind": "^1.1.2"
  },
  "engines": {
    "node": ">= 0.4"
  }
},
"node_modules/call-bound": {
  "version": "1.0.3",
  "resolved": "https://registry.npmjs.org/call-bound/-/call-bound-1.0.3.tgz",
  "integrity": "sha512-YTd+6wGlNiPxSuri7Y6X8tY2dmm12UMH66RpKMhiX6rsk5wXXnYgbUcOt8kiS31/AjfoTOvCsE+w8nZQLQnzHA==",
  "dependencies": {
    "call-bind-apply-helpers": "^1.0.1",
    "get-intrinsic": "^1.2.6"
  },
},

```

```

"engines": {
  "node": ">= 0.4"
},
"funding": {
  "url": "https://github.com/sponsors/ljharb"
},
"node_modules/concat-stream": {
  "version": "1.6.2",
  "resolved": "https://registry.npmjs.org/concat-stream/-/concat-stream-1.6.2.tgz",
  "integrity": "sha512-27HBghJxjiZtIk3Ycvn/4kbJk/1uZuJFfuPEns6LaEvpvG1f0hTea8lilrouyo9mVc2GWdcEZ8OLOmSADlrCw==",
  "engines": [
    "node >= 0.8"
  ],
  "dependencies": {
    "buffer-from": "^1.0.0",
    "inherits": "^2.0.3",
    "readable-stream": "^2.2.2",
    "typedarray": "^0.0.6"
  }
},
"node_modules/content-disposition": {
  "version": "0.5.4",
  "resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.4.tgz",
  "integrity": "sha512-FveZTNuGw04cx1AiWbzi6zTAL/lhehaWbTtgluJh4/E95DqMwTmha3KZN1aAWA8cFIhHzMZUvLevw5Rqk+tsQ==",
  "dependencies": {
    "safe-buffer": "5.2.1"
  },
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/content-type": {
  "version": "1.0.5",
  "resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.5.tgz",
  "integrity": "sha512-nTjqfcBFEipKdXCv4YDQWCfmcLZKm81ldF0pAopTvyRFGVbcR6P/VAAd5G7N+0tTr8QqiU0tFadD6FK4NtJwOA==",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/cookie": {
  "version": "0.7.1",
  "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.1.tgz",
  "integrity": "sha512-6DnInpx7SJ2AK3+CTUE/ZM0vWTUboZCegxhC2xiIydHR9jNuTAASBrfEpHhIGOZw/nX51bHt6YQl8jsGo4y/0w==",
  "engines": {
    "node": ">= 0.6"
  }
}

```

```

},
"node_modules/cookie-signature": {
  "version": "1.0.6",
  "resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",
  "integrity": "sha512-
QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQIrUgOWtHdNDtPiywmFbiS12VjotIXLrK
M3orQ=="
},
"node_modules/core-util-is": {
  "version": "1.0.3",
  "resolved": "https://registry.npmjs.org/core-util-is/-/core-util-is-1.0.3.tgz",
  "integrity": "sha512-
ZQBvi1DcpJ4GDqanjucZ2Hj3wEO5pZDS89BWbkcrvdxksJorwUDDZamX9ldFkp9aw2lmBDLgkObEA4
DWNJ9FYQ=="
},
"node_modules/cors": {
  "version": "2.8.5",
  "resolved": "https://registry.npmjs.org/cors/-/cors-2.8.5.tgz",
  "integrity": "sha512-
KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVd1ng7zOHx+YrEfInLG7q4n6GHQ9cDtxv/
P6g==",
  "dependencies": {
    "object-assign": "^4",
    "vary": "^1"
  },
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/debug": {
  "version": "2.6.9",
  "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
  "integrity": "sha512-
bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcK
GsHMA==",
  "dependencies": {
    "ms": "2.0.0"
  }
},
"node_modules/depd": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",
  "integrity": "sha512-
g7nH6P6dyDioJogAAGprGpCtVImJhpPk/roCzdb3flh61/s/nPsfR6onyMwkCAR/OlC3yBC0IESvUoQEAss
Irw==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/destroy": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/destroy/-/destroy-1.2.0.tgz",
  "integrity": "sha512-
2sJGJTaXIIaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxBpweaVcPevC2l3KpjYCx3Ny
pQQgaJg==",

```

```

    "engines": {
      "node": ">= 0.8",
      "npm": "1.2.8000 || >= 1.4.16"
    }
  },
  "node_modules/dotenv": {
    "version": "16.4.7",
    "resolved": "https://registry.npmjs.org/dotenv/-/dotenv-16.4.7.tgz",
    "integrity": "sha512-47qPchRCykZC03FhkYAhrrvwU4xDBFIj1QPqaarj6mdM/hgUzfPHcpkHJOn3mJAufFeeAxAzGsr5X0M4k6fLZQ==",
    "engines": {
      "node": ">=12"
    },
    "funding": {
      "url": "https://dotenvx.com"
    }
  },
  "node_modules/dunder-proto": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/dunder-proto/-/dunder-proto-1.0.1.tgz",
    "integrity": "sha512-KIN/nDJBQRcXw0MLVhZE9iQHmG68qAVIBg9CqmUYjmQlhgi9U5MFvrqkUL5FbtyyzZuOeOt0zdeRe4UY7ct+A==",
    "dependencies": {
      "call-bind-apply-helpers": "^1.0.1",
      "es-errors": "^1.3.0",
      "gopd": "^1.2.0"
    },
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/ecdsa-sig-formatter": {
    "version": "1.0.11",
    "resolved": "https://registry.npmjs.org/ecdsa-sig-formatter/-/ecdsa-sig-formatter-1.0.11.tgz",
    "integrity": "sha512-nagl3RYrbNv6kQkeJIpt6NJZy8twLB/2vtz6yN9Z4vRKHN4/QZJIEbqohALSgwKdnksuY3k5Addp5lg8sVoVcQ==",
    "dependencies": {
      "safe-buffer": "^5.0.1"
    }
  },
  "node_modules/ee-first": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",
    "integrity": "sha512-WMwm9LhRUo+WUaRN+vRuETqG89IgzPhVSNkdFgeb6sS/E4OrDIN7t48CAewSHXc6C8leFD8KKfr5vY61brQlow=="
  },
  "node_modules/encodeurl": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",
    "integrity": "sha512-1f901113W17e381R3eQ16e24u0f0eRU094yK7p6oIo587F7v8eF0AaB47Y1f4866h8a8/8rZvaygz2R0Q0n9HRI4m6JuGIV1eFlmvJB7ZEVxu93IrMyiMsGC0lrMJMWzRgx6WGquyfQgZVb31vhGgXnfmPNNX

```

```

mxnOkRBr=="",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/engine.io": {
  "version": "6.6.4",
  "resolved": "https://registry.npmjs.org/engine.io/-/engine.io-6.6.4.tgz",
  "integrity": "sha512-ZCkljSYNDyGn0R6ewHDtXgns/Zre/NT6Agvq1/WobF7JXgFff4SeDroKiCO3fNJreU9YG429Sc81o4w5ok/W5g==",
  "dependencies": {
    "@types/cors": "^2.8.12",
    "@types/node": ">=10.0.0",
    "accepts": "~1.3.4",
    "base64id": "2.0.0",
    "cookie": "~0.7.2",
    "cors": "~2.8.5",
    "debug": "~4.3.1",
    "engine.io-parser": "~5.2.1",
    "ws": "~8.17.1"
  },
  "engines": {
    "node": ">=10.2.0"
  }
},
"node_modules/engine.io-parser": {
  "version": "5.2.3",
  "resolved": "https://registry.npmjs.org/engine.io-parser/-/engine.io-parser-5.2.3.tgz",
  "integrity": "sha512-HqD3yTBfnBxIrbnM1DoD6Pcq8NECnh8d4As1Qgh0z5Gg3jRRIqijury0CL3ghu/edArpUYiYqQiDUQBIs4np3Q==",
  "engines": {
    "node": ">=10.0.0"
  }
},
"node_modules/engine.io/node_modules/cookie": {
  "version": "0.7.2",
  "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.2.tgz",
  "integrity": "sha512-yki5XnKuf750l50uGTllt6kKILY4nQ1eNIQatoXEBYZ5dWgnKqbnqmTrBE5B4N7lrMJKQ2ytWMiTO2o0v6Ew/w==",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/engine.io/node_modules/debug": {
  "version": "4.3.7",
  "resolved": "https://registry.npmjs.org/debug/-/debug-4.3.7.tgz",
  "integrity": "sha512-Er2nc/H7RrMXZBFCEim6TCmMk02Z8vLC2Rbi1KEBggpo0fS6l0S1nnapwmiI3yW/+GOJap1Krg4w0Hg80oCqgQ==",
  "dependencies": {
    "ms": "^2.1.3"
  },

```



```

    "engines": {
      "node": ">=6.0"
    },
    "peerDependenciesMeta": {
      "supports-color": {
        "optional": true
      }
    }
  },
  "node_modules/engine.io/node_modules/ms": {
    "version": "2.1.3",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
    "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjqz5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/es-define-property": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.1.tgz",
    "integrity": "sha512-eZnQl7xg6Lx5N48t1vW3l88QlT6G2WdYvFY0mzsktne6H28Z1k030G2rPb18l80Bg0jLKnQqS6tRp1ntY=",
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/es-errors": {
    "version": "1.3.0",
    "resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",
    "integrity": "sha512-U79f3t49945QwvSNvAAwL80qvJN3hvBz6A8b3vsf0Hi0qdGQ8a9CnFvIQ2UW40JpY4zV0HqAHq28X0G1Ug=",
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/es-object-atoms": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/es-object-atoms/-/es-object-atoms-1.1.1.tgz",
    "integrity": "sha512-FGgH2h8zKNim9ljj7dankFPcICIK9Cp5bm+c2gQSYePhpaG5+esrLODihIorn+Pe6FGJzWhXQotPv73jTal dXA==",
    "dependencies": {
      "es-errors": "^1.3.0"
    },
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/escape-html": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",
    "integrity": "sha512-DiNxw30kYRov7eS8aYb0XqT3994JNhf516VtMoY0ai0VmF+DqZ4jFRXWbDNRGE01Vpm0Wdhbk7A7QW7VtA="
  },
  "node_modules/etag": {
    "version": "1.8.1",
    "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
    "integrity": "sha512-aEL5g4iTKexCTIwO1p1m0nJabz5qQdZ5KQYd/6XhyHLU1OXsZGJgGMUgDkYhN7T5HIEoF4/t50G6QZgu5tA="
  },
  "node_modules/express": {
    "version": "4.18.2",
    "resolved": "https://registry.npmjs.org/express/-/express-4.18.2.tgz",
    "integrity": "sha512-5/P0L3A5+LZA6JJZ7Cjw5S5rky/wwV1DHxWiy6oLizLlN5SEGVgzZdKX0Lndmqhrpg2nD+ghaWbWJq1uKYEcg=",
    "dependencies": {
      "accepts": "~1.3.8",
      "array-flatten": "1.1.0",
      "body-parser": "1.20.1",
      "content-disposition": "0.5.4",
      "content-type": "~1.0.4",
      "cookie": "0.5.0",
      "cookie-signature": "1.0.6",
      "debug": "2.6.9",
      "depd": "2.0.0",
      "encodeurl": "~1.0.2",
      "escape-html": "~1.0.3",
      "etag": "~1.8.1",
      "finalhandler": "1.2.0",
      "fresh": "0.5.2",
      "http-errors": "2.0.0",
      "merge-descriptors": "1.0.1",
      "methods": "~1.1.2",
      "mime": "2.6.0",
      "morgan": "1.10.0",
      "multer": "1.4.4-lts",
      "on-finished": "2.4.1",
      "parseurl": "~1.3.3",
      "path-to-regexp": "0.1.7",
      "proxy-addr": "~2.0.7",
      "qs": "6.10.3",
      "range-parser": "~1.2.1",
      "raw-body": "2.5.1",
      "router": "1.3.3",
      "safe-buffer": "5.2.1",
      "send": "0.18.0",
      "serve-static": "1.15.0",
      "setprototypeof": "1.2.0",
      "statuses": "1.5.0",
      "toidentifier": "1.0.1",
      "type-is": "~1.6.18",
      "unpipe": "1.0.0",
      "utils-merge": "1.0.1",
      "vary": "1.1.2"
    },
    "engines": {
      "node": ">= 0.10.0"
    }
  },
  "node_modules/fresh": {
    "version": "0.5.2",
    "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",
    "integrity": "sha512-zJywOo1md+W36UYOGdY5UTtEylfj9Ap6jVW9ajqg8+sGtEzaj67jFoQpYfkg97qHJcVwf+7/eRbH2BOMatUdw="
  },
  "node_modules/fn-name": {
    "version": "1.0.4",
    "resolved": "https://registry.npmjs.org/fn-name/-/fn-name-1.0.4.tgz",
    "integrity": "sha512-2811W364G33ZD50833330YU4q39+3Y6ZM44LH3z6S5yJ0hgB9ab3xlsck6RZu6395r8E4Yb9P4m+U88g=",
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/forwarded": {
    "version": "0.2.0",
    "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",
    "integrity": "sha512-buRG0fpBt6hkWbmGgF9BxD17oO/9U5Ua32j5C1877O86sHtrTbpVZP303uIk5k1+ynZSMYyWzvvuiDQQ6AA="
  },
  "node_modules/fresh-0.3.0": {
    "version": "0.3.0",
    "resolved": "https://registry.npmjs.org/fresh-0.3.0.tgz",
    "integrity": "sha512-0J6v8S4BQytdy9D/OnS30hBwNjxWuSDE4wqTQcP39eFm6wUyXABPrAqzoMK/0LhL9HnG5FCe69ZHFQJog6A="
  },
  "node_modules/fresh-0.5.2": {
    "version": "0.5.2",
    "resolved": "https://registry.npmjs.org/fresh-0.5.2.tgz",
    "integrity": "sha512-tJhJc+sBjMx6Wq10GztBZQ6Y4uF4eT75QK3xHY2avbCkXkGQm/PHU0tFb2tX6A4XWKI6FLHkU12GZZzUk3A="
  },
  "node_modules/fresh-1.0.0": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/fresh-1.0.0.tgz",
    "integrity": "sha512-RDk1B4E846gK++8iQWbJG++6RYdG4LW2w1X3IAZK3894aJv+rDzqrz4oHqCY3WhR2SU2x6L7HDw/Xq748g="
  },
  "node_modules/fresh-2.0.0": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/fresh-2.0.0.tgz",
    "integrity": "sha512-Rt/9VqEkW/6Ezd9w0E18oQR15o4u0V7IxyEhcKyz8iZQlP5H0b35uXksqkQZ+pQX6oNlsG2NtkYXN52g8A="
  },
  "node_modules/fresh-3.0.0": {
    "version": "3.0.0",
    "resolved": "https://registry.npmjs.org/fresh-3.0.0.tgz",
    "integrity": "sha512-3L0F3L4qD823Q444E89Mbm3330bWRQ6qByD633zc9WUO411v6jkBojW2ZJxZB8mL7QqdFkUgROO0tUzYw="
  },
  "node_modules/fresh-4.0.0": {
    "version": "4.0.0",
    "resolved": "https://registry.npmjs.org/fresh-4.0.0.tgz",
    "integrity": "sha512-RpKpZCz1hZl5Jt0LqRwFJF7u4D6CZ3FmZ8d8eZuWYq1QsT3A4Hn0pZdcGg31gS/H1lbshdZfJq841g=="
  },
  "node_modules/fresh-5.0.0": {
    "version": "5.0.0",
    "resolved": "https://registry.npmjs.org/fresh-5.0.0.tgz",
    "integrity": "sha512-0t4rLl1fBZvYFTrzZS2WaECBkLJZWcfs4YnR0tqiy612OJgY5fwWCvRr6MHmNSBZKuUm514Yf5p5qQ65o8A="
  },
  "node_modules/fresh-6.0.0": {
    "version": "6.0.0",
    "resolved": "https://registry.npmjs.org/fresh-6.0.0.tgz",
    "integrity": "sha512-4n3KkWj5X1WPcG2FpeoRWlGz4PqlLoj04gqyrw1zRZtDnYp8qS60HjsXZjx04WlRi6RdK7BpEaRIFTuYUw="
  },
  "node_modules/fresh-7.0.0": {
    "version": "7.0.0",
    "resolved": "https://registry.npmjs.org/fresh-7.0.0.tgz",
    "integrity": "sha512-5P0zG5lS+7oN5DlGeybKwqA5TfZjQJ6E+Z4izy7v8TAp6PJ5WbZuW5GyXzdiBc2aP975aLWZtVjKq33tA="
  },
  "node_modules/fresh-8.0.0": {
    "version": "8.0.0",
    "resolved": "https://registry.npmjs.org/fresh-8.0.0.tgz",
    "integrity": "sha512-0B7fy3VdX219p1Fz9u0ZpUWV4d8B7bXU3QbEPNz82T2dwyW5Gza2CQvW5QTu0Ll15+lXU0oT8IP0vPw=="
  },
  "node_modules/fresh-9.0.0": {
    "version": "9.0.0",
    "resolved": "https://registry.npmjs.org/fresh-9.0.0.tgz",
    "integrity": "sha512-2F3yFp1kRCPyJH0EiNoFgKhD0wvD0I3NL1EjyL784umLcAZdP8pBdI4bK2A9Qbk39GdCzL61dL8vDk+VEg="
  },
  "node_modules/fresh-10.0.0": {
    "version": "10.0.0",
    "resolved": "https://registry.npmjs.org/fresh-10.0.0.tgz",
    "integrity": "sha512-8cd8LsfEBZUwFYmeBguI2jJfNw2ndHA9o8guYvAZo4taCFxpcR7ZjqzVYFaz2fJQTzuktqrV2OFbIXG7KuA=="
  },
  "node_modules/fresh-11.0.0": {
    "version": "11.0.0",
    "resolved": "https://registry.npmjs.org/fresh-11.0.0.tgz",
    "integrity": "sha512-6F0WgJ1E7y1wAp0s+sCreZTbDrFqLw0lmjIbTl9Jpe5rJ8VGxMEqW4lZX68F389n7GZFbjizE9WZ1YUa3A=="
  },
  "node_modules/fresh-12.0.0": {
    "version": "12.0.0",
    "resolved": "https://registry.npmjs.org/fresh-12.0.0.tgz",
    "integrity": "sha512-gXwOGtNn8/Jk2C2R/Ft0aGpsW6JO6V74BPk0A8Nchl5HlWsw5qGfXWw19B91zF1Za1oYDkZM14q1w6MGw=="
  },
  "node_modules/fresh-13.0.0": {
    "version": "13.0.0",
    "resolved": "https://registry.npmjs.org/fresh-13.0.0.tgz",
    "integrity": "sha512
```

```

},
"node_modules/etag": {
  "version": "1.8.1",
  "resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",
  "integrity": "sha512-aIL5Fx7mawVa300a12BnEE4iNvo1qETxLrPI/o05L7z6go7fCw1J6EQmbK4FmJ2AS7kgVF/KEZWufBfdC
lMcPg==",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/express": {
  "version": "4.21.2",
  "resolved": "https://registry.npmjs.org/express/-/express-4.21.2.tgz",
  "integrity": "sha512-28HqgMZAmyh1Czt9ny7qr6ek2qddF4FclbMzwhCREB6OFfH+rXAnuNCwo1/wFvrtbgsQDb4kSbX9de9lF
brXnA==",
  "dependencies": {
    "accepts": "~1.3.8",
    "array-flatten": "1.1.1",
    "body-parser": "1.20.3",
    "content-disposition": "0.5.4",
    "content-type": "~1.0.4",
    "cookie": "0.7.1",
    "cookie-signature": "1.0.6",
    "debug": "2.6.9",
    "depd": "2.0.0",
    "encodeurl": "~2.0.0",
    "escape-html": "~1.0.3",
    "etag": "~1.8.1",
    "finalhandler": "1.3.1",
    "fresh": "0.5.2",
    "http-errors": "2.0.0",
    "merge-descriptors": "1.0.3",
    "methods": "~1.1.2",
    "on-finished": "2.4.1",
    "parseurl": "~1.3.3",
    "path-to-regexp": "0.1.12",
    "proxy-addr": "~2.0.7",
    "qs": "6.13.0",
    "range-parser": "~1.2.1",
    "safe-buffer": "5.2.1",
    "send": "0.19.0",
    "serve-static": "1.16.2",
    "setprototypeof": "1.2.0",
    "statuses": "2.0.1",
    "type-is": "~1.6.18",
    "utils-merge": "1.0.1",
    "vary": "~1.1.2"
  },
  "engines": {
    "node": ">= 0.10.0"
  },
  "funding": {
    "type": "opencollective",

```

```

    "url": "https://opencollective.com/express"
  },
  "node_modules/finalhandler": {
    "version": "1.3.1",
    "resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.3.1.tgz",
    "integrity": "sha512-6BN9trH7bp3qvnRyZsBz+g3lZxTNZTbVO2EV1CS0WlCdbawYVdYvGfIME/9QP0h0pYlCDBCTjYa9nZzMDpyxQ==",
    "dependencies": {
      "debug": "2.6.9",
      "encodeurl": "~2.0.0",
      "escape-html": "~1.0.3",
      "on-finished": "2.4.1",
      "parseurl": "~1.3.3",
      "statuses": "2.0.1",
      "unpipe": "~1.0.0"
    },
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/forwarded": {
    "version": "0.2.0",
    "resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",
    "integrity": "sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyablzMzOA5z9xA9L1KNjk/Nt6MT9aYow==",
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/fresh": {
    "version": "0.5.2",
    "resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",
    "integrity": "sha512-t7RK96H8/qV8zT16/zt1154T31I1zQeWlE206Vl0uWl1K0ZL0f8Hn3B41Ea17gO8Df12a0iZs7H0G1Q==",
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/function-bind": {
    "version": "1.1.2",
    "resolved": "https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz",
    "integrity": "sha512-7P80w905408D9C56DwBtEBYgfO7RDjQuWgkCknt15iR+Yvu01k9N8qJbUM7bRnRgNZuYyJBBWx7oizI0Q==",
    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/get-intrinsic": {
    "version": "1.2.7",
    "resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.2.7.tgz",

```

```

    "integrity": "sha512-
VW6Pxhsrk0KAOqs3WE d0kIDiF/+V7gQOpAvY1jVU/LHmaD/kQO4523aiJuikX/QAKYiW6x8Jh+RJej1a
lmdtCA==",
    "dependencies": {
      "call-bind-apply-helpers": "^1.0.1",
      "es-define-property": "^1.0.1",
      "es-errors": "^1.3.0",
      "es-object-atoms": "^1.0.0",
      "function-bind": "^1.1.2",
      "get-proto": "^1.0.0",
      "gopd": "^1.2.0",
      "has-symbols": "^1.1.0",
      "hasown": "^2.0.2",
      "math-intrinsics": "^1.1.0"
    },
    "engines": {
      "node": ">= 0.4"
    },
    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/get-proto": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/get-proto/-/get-proto-1.0.1.tgz",
    "integrity": "sha512-
sTSfBjoXBp89JvIKIefqw7U2CCebsc74kiY6awiGogKtoSGbgjYE/G/+l9sF3MWFPNc9IcoOC4ODfKHfxF
mp0g==",
    "dependencies": {
      "dunder-proto": "^1.0.1",
      "es-object-atoms": "^1.0.0"
    },
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/gopd": {
    "version": "1.2.0",
    "resolved": "https://registry.npmjs.org/gopd/-/gopd-1.2.0.tgz",
    "integrity": "sha512-
ZUKRh6/kUFoAiTA tTYPZJ3hw9wNxx+BIBOijnlg9PnrJsCcSjslwyD6vJpaYtgnzDrKYRSqf3OO6Rfa9
3xsRg==",
    "engines": {
      "node": ">= 0.4"
    },
    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/has-symbols": {
    "version": "1.1.0",
    "resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.1.0.tgz",
    "integrity": "sha512-
1cDNdwJ2Jaohmb3sg4OmKaMBwuC48sYni5HUw2DvsC8LjGTLK9h+eb1X6RyuOHe4hT0ULCW68iom
hjUoKUqlPQ==",

```

```

"engines": {
  "node": ">= 0.4"
},
"funding": {
  "url": "https://github.com/sponsors/ljharb"
},
"node_modules/hasown": {
  "version": "2.0.2",
  "resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz",
  "integrity": "sha512-0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cII9y+CyS8rG3nL48Bclra9HmKhVUUHyPWNH5Y7xDwAB7bfgSjkUMQ==",
  "dependencies": {
    "function-bind": "^1.1.2"
  },
  "engines": {
    "node": ">= 0.4"
  }
},
"node_modules/http-errors": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-2.0.0.tgz",
  "integrity": "sha512-0w00uSjf0A1tU0qK1tZjW6dJHvQz3X1oFv1R4YjNU13o0u9tWvD9kVgR6o4q4WV7e6/2ZpoL7do7LwUw==",
  "dependencies": {
    "depd": "2.0.0",
    "inherits": "2.0.4",
    "setprototypeof": "1.2.0",
    "statuses": "2.0.1",
    "toidentifier": "1.0.1"
  },
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/iconv-lite": {
  "version": "0.4.24",
  "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",
  "integrity": "sha512-v3MXnZAcvnywkTUEZomIActle7RXXeedOR3lwwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1HwLI21bEqdpj8/rA==",
  "dependencies": {
    "safer-buffer": ">= 2.1.2 < 3"
  },
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/inherits": {
  "version": "2.0.4",
  "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",
  "integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSoIFeMpXKxa4

```

```
dYeZIQqewQ=="
},
"node_modules/ipaddr.js": {
  "version": "1.9.1",
  "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",
  "integrity": "sha512-0KI/607xoxStoH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZIXSHBYX
iYUPO3g==",
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/isarray": {
  "version": "1.0.0",
  "resolved": "https://registry.npmjs.org/isarray/-/isarray-1.0.0.tgz",
  "integrity": "sha512-0
VLghIWNM6ELQzo7zwmcg0NmTVyWKYjvIeM83yjp0wRDTmUnrM678fQbcKBo6n2CJEF0szoG//ytg+
TKla89ALQ==",
  "node_modules/json": {
    "version": "11.0.0",
    "resolved": "https://registry.npmjs.org/json/-/json-11.0.0.tgz",
    "integrity": "sha512-
N/ITv3Yw9Za8cGxuQqSqrq6RHnlaHWZkAFavcfpH/R52522c26EbihMxnY7A1chxfXJ4d+cEFIsyTgfi9Gi
hrA==",
    "bin": {
      "json": "lib/json.js"
    },
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/jsonwebtoken": {
    "version": "9.0.2",
    "resolved": "https://registry.npmjs.org/jsonwebtoken/-/jsonwebtoken-9.0.2.tgz",
    "integrity": "sha512-
PRp66vJ865SSqOlgqS8hujT5U4AOgMfhrwYIuIhfKaoSCZcirmASQr8CX7cUg+RMih+hgznrjp99o+W4p
JLHQ==",
    "dependencies": {
      "jws": "^3.2.2",
      "lodash.includes": "^4.3.0",
      "lodash.isboolean": "^3.0.3",
      "lodash.isinteger": "^4.0.4",
      "lodash.isnumber": "^3.0.3",
      "lodash.isplainobject": "^4.0.6",
      "lodash.isstring": "^4.0.1",
      "lodash.once": "^4.0.0",
      "ms": "^2.1.1",
      "semver": "^7.5.4"
    },
    "engines": {
      "node": ">=12",
      "npm": ">=6"
    }
  },
}
```

```

"node_modules/jsonwebtoken/node_modules/ms": {
  "version": "2.1.3",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
  "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjqz5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA=="
},
"node_modules/jwa": {
  "version": "1.4.1",
  "resolved": "https://registry.npmjs.org/jwa/-/jwa-1.4.1.tgz",
  "integrity": "sha512-qiLX/xhEEFKUAJ6FiBMbes3w9ATzyk5W7Hvzpa/SLYdxNtng+gcurvrI7TbACjIXlsJyr05/S1oUhZrc63evQA==",
  "dependencies": {
    "buffer-equal-constant-time": "1.0.1",
    "ecdsa-sig-formatter": "1.0.11",
    "safe-buffer": "^5.0.1"
  }
},
"node_modules/jws": {
  "version": "3.2.2",
  "resolved": "https://registry.npmjs.org/jws/-/jws-3.2.2.tgz",
  "integrity": "sha512-Edcng4rvS000QSiU3oGip6Clag+QvCt1pDLxKMsuLWvsyBBY2wXvIn6ptXh8dIYYk4d/iqZ19jLsQUdxZBk=",
  "dependencies": {
    "jwa": "^1.4.1",
    "safe-buffer": "^5.0.1"
  }
},
"node_modules/kareem": {
  "version": "2.6.3",
  "resolved": "https://registry.npmjs.org/kareem/-/kareem-2.6.3.tgz",
  "integrity": "sha512-68H08310435099308pU46766t0B0wWpB8Q4wbtY+GRhGabaB4xVJcW69lkpaSszGgQh7T6gKpTHgrWecA=",
  "engines": {
    "node": ">=12.0.0"
  }
},
"node_modules/lodash.includes": {
  "version": "4.3.0",
  "resolved": "https://registry.npmjs.org/lodash.includes/-/lodash.includes-4.3.0.tgz",
  "integrity": "sha512-6U1Wz6Bt+oqD6LkSY0F6Q5KwA87nN4SbT5F1jyL5LFf/EqGrJy6q7tFp78Kb8ocxNbkeI8a9Zq7HfH+A=",
  "node_modules/lodash.isboolean": {
    "version": "3.0.3",
    "resolved": "https://registry.npmjs.org/lodash.isboolean/-/lodash.isboolean-3.0.3.tgz",
    "integrity": "sha512-BL58GE44e85F4y1r6bbFkafVj9uANm/FNSWd5HkiB4WH6WVp+ZpjbGty/U3isSHlaQSKLct3x4jIDcS7jYzAw="
  },
  "node_modules/lodash.isinteger": {

```

```

    "version": "4.0.4",
    "resolved": "https://registry.npmjs.org/lodash.isinteger/-/lodash.isinteger-4.0.4.tgz",
    "integrity": "sha512-DBwtEWN2caHQ9/imiNeEA5ys1JoRtRfY3d7V9wkqtbycnAmTvRRmbHKDV4a0EYc678/dia0jrte4tjYwVBaZUA=="
  },
  "node_modules/lodash.isnumber": {
    "version": "3.0.3",
    "resolved": "https://registry.npmjs.org/lodash.isnumber/-/lodash.isnumber-3.0.3.tgz",
    "integrity": "sha512-QYqzpfwO3/CWf3XP+Z+tkQsfaLL/EnUIXWVklk5FUPc4sBdTehEqZONuyRt2P67PXAk+NXmTBcc97zw9t1lFQrw=="
  },
  "node_modules/lodash.isplainobject": {
    "version": "4.0.6",
    "resolved": "https://registry.npmjs.org/lodash.isplainobject/-/lodash.isplainobject-4.0.6.tgz",
    "integrity": "sha512-oSXzaWypCMHkPC3NvBEaPHf0KsA5mvPrOPgQWDsbg8n7orZ290M0BmC/jgRZ4vcJ6DTA hjrsSY gdsW/F+MFOBA=="
  },
  "node_modules/lodash.isstring": {
    "version": "4.0.1",
    "resolved": "https://registry.npmjs.org/lodash.isstring/-/lodash.isstring-4.0.1.tgz",
    "integrity": "sha512-0wJxfxH1wgO3GrbuP+dTTk7op+6L41QCXbGInEmD+ny/G/eCqGzxyCsh7159S+mgDDcoarnBw6PC1PS5+wUGgw=="
  },
  "node_modules/lodash.once": {
    "version": "4.1.1",
    "resolved": "https://registry.npmjs.org/lodash.once/-/lodash.once-4.1.1.tgz",
    "integrity": "sha512-Sb487aTOCr9drQVL8pIxOzVhafOjZN9UU54hiN8PU3uAiSV7lx1yYNpbNmex2PK6dSJoNTSJUUswT651yww3Mg=="
  },
  "node_modules/math-intrinsics": {
    "version": "1.1.0",
    "resolved": "https://registry.npmjs.org/math-intrinsics/-/math-intrinsics-1.1.0.tgz",
    "integrity": "sha512-/IXtbwEk5HTPyEwyKX6hGkYXxM9nbj64B+ilVJnC/R6B0pH5G4V3b0pVbL7DBj4tkhBAppbQUlf6F6Xl9LHu1g==",
    "engines": {
      "node": ">= 0.4"
    }
  },
  "node_modules/media-typer": {
    "version": "0.3.0",
    "resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",
    "integrity": "sha512-dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJYmUmknUF5HwMLOgb5O+a3KxfWapPQ==",
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/memory-pager": {

```



```

    "version": "1.5.0",
    "resolved": "https://registry.npmjs.org/memory-pager/-/memory-pager-1.5.0.tgz",
    "integrity": "sha512-ZS4Bp4r/Zoeq6+NLJpP+0Zzm0pR8whtGPf1XExKLJBaczGMnSi3It14OiNCSjtjQjM6NU1okjQGSxgEZN8eBYKg==",
  },
  "node_modules/merge-descriptors": {
    "version": "1.0.3",
    "resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.3.tgz",
    "integrity": "sha512-gaNvAS7TZ897/rVaZ0nMtAyxNyi/pdbjbAwUpFQpN70GqnVfOiXpeUUMKRBmzXaSQ8DdTX4/0ms62r2K+hE6mQ==",
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/methods": {
    "version": "1.1.2",
    "resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",
    "integrity": "sha512-iclAHeNqNm68zFtnZ0e+1L2yUIdivzNoauKU4WBA3VvH/vPFieF7qfRlwUZU+DA9P9bPXIS90ulxoUoCH23sV2w==",
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/mime": {
    "version": "1.6.0",
    "resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",
    "integrity": "sha512-x0Vn8spl+wuJlO6S7gnbaQg8Pxh4NNHb7KSINmEWKiPE4RKOpIvijn+NkmYmmRgP68mc70j2EbeTFRsrswaQeg==",
    "bin": {
      "mime": "cli.js"
    },
    "engines": {
      "node": ">=4"
    }
  },
  "node_modules/mime-db": {
    "version": "1.52.0",
    "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",
    "integrity": "sha512-sPU4uV7dY1vtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==",
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/mime-types": {
    "version": "2.1.35",
    "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",
    "integrity": "sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==",

```

```

"dependencies": {
  "mime-db": "1.52.0"
},
"engines": {
  "node": ">= 0.6"
},
"node_modules/minimist": {
  "version": "1.2.8",
  "resolved": "https://registry.npmjs.org/minimist/-/minimist-1.2.8.tgz",
  "integrity": "sha512-2yyAR8qBkN3YuheJanUpWC5U3bb5osDywNB8RzDVIDwDHBocAJveqqj1u8+SVD7jkWT4yvsHCpWq
qWqAxb0zCA==",
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/mkdirp": {
  "version": "0.5.6",
  "resolved": "https://registry.npmjs.org/mkdirp/-/mkdirp-0.5.6.tgz",
  "integrity": "sha512-FP+p8RB8OWpF3YZBCrP5gtADmtXApB5AMLn+vdyA+PyxCjrCs00mjoyUozssO33cwDeT3wNGdLxJ5
M//YqtHAJw==",
  "dependencies": {
    "minimist": "^1.2.6"
  },
  "bin": {
    "mkdirp": "bin/cmd.js"
  }
},
"node_modules/mongodb": {
  "version": "6.12.0",
  "resolved": "https://registry.npmjs.org/mongodb/-/mongodb-6.12.0.tgz",
  "integrity": "sha512-
RM7AHLvYfs7jv7+BXund/kR64DryVI+cHbVAy9P61fnb1RcWZqOW1/Wj2YhqMCx+MuYhqTRGv7Aw
HBzmsCKBfA==",
  "dependencies": {
    "@mongodb-js/saslprep": "^1.1.9",
    "bson": "^6.10.1",
    "mongodb-connection-string-url": "^3.0.0"
  },
  "engines": {
    "node": ">=16.20.1"
  },
  "peerDependencies": {
    "@aws-sdk/credential-providers": "^3.188.0",
    "@mongodb-js/zstd": "^1.1.0 || ^2.0.0",
    "gcp-metadata": "^5.2.0",
    "kerberos": "^2.0.1",
    "mongodb-client-encryption": ">=6.0.0 <7",
    "snappy": "^7.2.2",
    "socks": "^2.7.1"
  },
  "peerDependenciesMeta": {
    "@aws-sdk/credential-providers": {

```

```

    "optional": true
  },
  "@mongodb-js/zstd": {
    "optional": true
  },
  "gcp-metadata": {
    "optional": true
  },
  "kerberos": {
    "optional": true
  },
  "mongodb-client-encryption": {
    "optional": true
  },
  "snappy": {
    "optional": true
  },
  "socks": {
    "optional": true
  }
},
"node_modules/mongodb-connection-string-url": {
  "version": "3.0.2",
  "resolved": "https://registry.npmjs.org/mongodb-connection-string-url/-/mongodb-connection-string-url-3.0.2.tgz",
  "integrity": "sha512-rMO7CGO/9BFwyZABcKAWL8UJwH/Kc2x0g72uhDWzG48URRax5TCIcJ7Rc3RZqffZzO/Gwff/jyKwC U9TN8gehA==",
  "dependencies": {
    "@types/whatwg-url": "^11.0.2",
    "whatwg-url": "^14.1.0 || ^13.0.0"
  }
},
"node_modules/mongoose": {
  "version": "8.9.7",
  "resolved": "https://registry.npmjs.org/mongoose/-/mongoose-8.9.7.tgz",
  "integrity": "sha512-mvNXmU0V8qZzMR/qoK2mjT4Ti2ALdtfS0teK+twxhlGkwzOD76V02/zWajTu2MJ7QyEmZe9OWvnJsI Y0iAuX3Q==",
  "dependencies": {
    "bson": "^6.10.1",
    "kareem": "2.6.3",
    "mongodb": "~6.12.0",
    "mpath": "0.9.0",
    "mquery": "5.0.0",
    "ms": "2.1.3",
    "sift": "17.1.3"
  },
  "engines": {
    "node": ">=16.20.1"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/mongoose"
  }
}

```

```

    }
  },
  "node_modules/mongoose/node_modules/ms": {
    "version": "2.1.3",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
    "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhJzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
  },
  "node_modules/mpath": {
    "version": "0.9.0",
    "resolved": "https://registry.npmjs.org/mpath/-/mpath-0.9.0.tgz",
    "integrity": "sha512-ikJRQTk8hw5DEoFVxHG1Gn9T/xcjtdnOKIU1JTmGjZZlg9LST2mBLmcX3/IClbgJydT2GOc15RnNy5mHmzfSew==",
    "engines": {
      "node": ">=4.0.0"
    }
  },
  "node_modules/mquery": {
    "version": "5.0.0",
    "resolved": "https://registry.npmjs.org/mquery/-/mquery-5.0.0.tgz",
    "integrity": "sha512-iQMncpmEK8R8ncT8HJGsGc9Dsp8xcgYMVSBs5jgnm1lFHTZqMJTUWTDx1LBO8+mK3tPNZWFLBg hQEIOULSTHZg==",
    "dependencies": {
      "debug": "4.x"
    },
    "engines": {
      "node": ">=14.0.0"
    }
  },
  "node_modules/mquery/node_modules/debug": {
    "version": "4.4.0",
    "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.0.tgz",
    "integrity": "sha512-6WTZ/IxCY/T6BALoZHaE4ctp9xm+Z5kY/pzYaCHRFeyVhojxlrM+46y68HA6hr0TcwEssoxNiDEUJQjFP Z/RYA==",
    "dependencies": {
      "ms": "^2.1.3"
    },
    "engines": {
      "node": ">=6.0"
    },
    "peerDependenciesMeta": {
      "supports-color": {
        "optional": true
      }
    }
  },
  "node_modules/mquery/node_modules/ms": {
    "version": "2.1.3",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
    "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhJzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3W

```

```

RTIA=="
  },
  "node_modules/ms": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
    "integrity": "sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05
A=="
  },
  "node_modules/multer": {
    "version": "1.4.5-lts.1",
    "resolved": "https://registry.npmjs.org/multer/-/multer-1.4.5-lts.1.tgz",
    "integrity": "sha512-ywPWvcDMeH+z9gQq5qYHCCy+ethsk4goepZ45GLD63fOu0YcNecQxi64nDs3qluZB+murG3/D4dJ7+d
GctcCQQ==",
    "dependencies": {
      "append-field": "^1.0.0",
      "busboy": "^1.0.0",
      "concat-stream": "^1.5.2",
      "mkdirp": "^0.5.4",
      "object-assign": "^4.1.1",
      "type-is": "^1.6.4",
      "xtend": "^4.0.0"
    },
    "engines": {
      "node": ">= 6.0.0"
    }
  },
  "node_modules/negotiator": {
    "version": "0.6.3",
    "resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.3.tgz",
    "integrity": "sha512-+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGLQk/k5giakcKsuxCObBRu6DSm9opw/O6slWbJdgh
QM4bBg==",
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/object-assign": {
    "version": "4.1.1",
    "resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",
    "integrity": "sha512-+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGLQk/k5giakcKsuxCObBRu6DSm9opw/O6slWbJdgh
rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU
5hj+BcUg==",
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/object-inspect": {
    "version": "1.13.4",
    "resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-1.13.4.tgz",
    "integrity": "sha512-W67iLl4J2EXEGTbfeHCffrjDfitvLANg0UIX3wFUUSTx92KXRFegMHUVgSqE+vvhAbi4WqjGg9czysT
V2Epbew==",
    "engines": {

```

```

    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/on-finished": {
  "version": "2.4.1",
  "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.4.1.tgz",
  "integrity": "sha512-oVlzkg3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJJePWYF/CImuDr8Ch5+kg==",
  "dependencies": {
    "ee-first": "1.1.1"
  },
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/parseurl": {
  "version": "1.3.3",
  "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
  "integrity": "sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Syzg1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/path-to-regexp": {
  "version": "0.1.12",
  "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.12.tgz",
  "integrity": "sha512-RA1GjUVMNvYF XuqovrEqZoxXW5NUZqbWktYz/Tt7nXerk0LbLbLQmrsgdeOxV5SFHf0UDggjS/bSeOZwt1pmEQ==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/process-nextick-args": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/process-nextick-args/-/process-nextick-args-2.0.1.tgz",
  "integrity": "sha512-3ouUOpQhtgrbOa17J7+uxOTpITYWaGP7/AhoR3+A+/1e9skrzelGi/dXzEYyvbxbubEF6Wn2ypscTKiKJFFnlag==",
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/proxy-addr": {
  "version": "2.0.7",
  "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",
  "integrity": "sha512-30sJl2ZOH8M1ZP6g0h0PwM0G1Kb+NOmW6ZxXbzQU6WvGSw7rUcnhMCbRKJIcK5TOge0deSvF9pADTgkRh4=
  "dependencies": {
    "forwarded": "0.2.0",
    "ipaddr.js": "1.9.1"
  },
  "engines": {
    "node": ">= 0.10"
  }
}

```

```

    }
  },
  "node_modules/punycode": {
    "version": "2.3.1",
    "resolved": "https://registry.npmjs.org/punycode/-/punycode-2.3.1.tgz",
    "integrity": "sha512-
vYt7UD1U9Wg6138shLtLOvdAu+8DsC/ilFtEVHcH+wydcSpNE20AfSOduf6MkRFahL5FY7X1oU7nKV
ZFtfq8Fg==",
    "engines": {
      "node": ">=6"
    }
  },
  "node_modules/qs": {
    "version": "6.13.0",
    "resolved": "https://registry.npmjs.org/qs/-/qs-6.13.0.tgz",
    "integrity": "sha512-
+38qI9SOOr8tfZ4QmJNpIMUxqjbe7LKvvZgWdExBOmd+egZTtjLB67Gu0HRX3u/XOq7UU2Nx6nsjvS16
Z9uwfpg==",
    "dependencies": {
      "side-channel": "^1.0.6"
    },
    "engines": {
      "node": ">=0.6"
    },
    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/range-parser": {
    "version": "1.2.1",
    "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",
    "integrity": "sha512-
Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRozfG3ccAveqCBwr/2yxQ5BVd/GTl
5agOwSg==",
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/raw-body": {
    "version": "2.5.2",
    "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.2.tgz",
    "integrity": "sha512-
8zGqypfENjCIqGhgXToc8aB2r7YrBX+AQAFIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+a48lt8qsCzirQ6loC
VfA==",
    "dependencies": {
      "bytes": "3.1.2",
      "http-errors": "2.0.0",
      "iconv-lite": "0.4.24",
      "unpipe": "1.0.0"
    },
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/readable-stream": {

```

```

    "version": "2.3.8",
    "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
    "integrity": "sha512-8p0AUk4XODgIewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5lRrFXGg71L15KG8QrPm
um45RTtdA==",
    "dependencies": {
      "core-util-is": "~1.0.0",
      "inherits": "~2.0.3",
      "isarray": "~1.0.0",
      "process-nextick-args": "~2.0.0",
      "safe-buffer": "~5.1.1",
      "string_decoder": "~1.1.1",
      "util-deprecate": "~1.0.1"
    }
  },
  "node_modules/readable-stream/node_modules/safe-buffer": {
    "version": "5.1.2",
    "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
    "integrity": "sha512-Gd2UZBJDkXlY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpScSmMJSOfF8LOIrt8ud/wPtojs
s4G6+g=="
  },
  "node_modules/safe-buffer": {
    "version": "5.2.1",
    "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
    "integrity": "sha512-rp3So07KcdmmKbGvgaNxQSiJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuC
XmpuYvmCXQ==",
    "funding": [
      {
        "type": "github",
        "url": "https://github.com/sponsors/feross"
      },
      {
        "type": "patreon",
        "url": "https://www.patreon.com/feross"
      },
      {
        "type": "consulting",
        "url": "https://feross.org/support"
      }
    ]
  },
  "node_modules/safer-buffer": {
    "version": "2.1.2",
    "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",
    "integrity": "sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqc
QriUtg=="
  },
  "node_modules/semver": {
    "version": "7.7.1",
    "resolved": "https://registry.npmjs.org/semver/-/semver-7.7.1.tgz",
    "integrity": "sha512-
hlq8tAfn0m/61p4BVRcPzIGr6LKIMwo4VM6dGi6pt4qcRkmNzTcWq6eCEjEh+qXjkMDvPIOFFSGwQjo

```



```

Ea6gyMA==",
  "bin": {
    "semver": "bin/semver.js"
  },
  "engines": {
    "node": ">=10"
  }
},
"node_modules/send": {
  "version": "0.19.0",
  "resolved": "https://registry.npmjs.org/send/-/send-0.19.0.tgz",
  "integrity": "sha512-dW41u5VfLXu8S.Jh5bwRmyYUbaAoSB3c9uQh6L8h/KtsFREPWpbX11rlJJo186Jc4nmci/sGUZ9a0a0J2zgfgq
2hw==",
  "dependencies": {
    "debug": "2.6.9",
    "depd": "2.0.0",
    "destroy": "1.2.0",
    "encodeurl": "~1.0.2",
    "escape-html": "~1.0.3",
    "etag": "~1.8.1",
    "fresh": "0.5.2",
    "http-errors": "2.0.0",
    "mime": "1.6.0",
    "ms": "2.1.3",
    "on-finished": "2.4.1",
    "range-parser": "~1.2.1",
    "statuses": "2.0.1"
  },
  "engines": {
    "node": ">= 0.8.0"
  }
},
"node_modules/send/node_modules/encodeurl": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",
  "integrity": "sha512-TPJXq8JqFaVYm2CWmPvnP2Iyo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/rTDf+rut/saiDiQEeVTNgAm
JEAOx0w==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/send/node_modules/ms": {
  "version": "2.1.3",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
  "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3W
RTIA=="
},
"node_modules/serve-static": {
  "version": "1.16.2",
  "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.16.2.tgz",
  "integrity": "sha512-VqpjJZKadQB/PEbEwvFdO43Ax5dFBZ2UECsz8bQ7pi7wt//PWelP6MN7eCnjsatYtBT6EuiClbjSWP2W
VqpjJZKadQB/PEbEwvFdO43Ax5dFBZ2UECsz8bQ7pi7wt//PWelP6MN7eCnjsatYtBT6EuiClbjSWP2W

```

```

rloTw=="",
  "dependencies": {
    "encodeurl": "~2.0.0",
    "escape-html": "~1.0.3",
    "parseurl": "~1.3.3",
    "send": "0.19.0"
  },
  "engines": {
    "node": ">= 0.8.0"
  }
},
"node_modules/setprototypeof": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",
  "integrity": "sha512-8580Z35970EHCuOwz24Hjuo1u9tm4894249aAcu1WkE18gQ1DWOsEeZpX1Dk2q1VbWSPbZLXGizkYqU8wUg==",
  "dependencies": {
    "es-errors": "^1.3.0",
    "object-inspect": "^1.13.3",
    "side-channel-list": "^1.0.0",
    "side-channel-map": "^1.0.1",
    "side-channel-weakmap": "^1.0.2"
  },
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/side-channel-list": {
  "version": "1.0.0",
  "resolved": "https://registry.npmjs.org/side-channel-list/-/side-channel-list-1.0.0.tgz",
  "integrity": "sha512-FCLHtRD/gnpCiCHEiJLOWdmFP+wzCmDEkc9y7NsYxeF4u7Btsn1ZuwgwJGxImImHicJArLP4R0yX4c2KCrMrTA==",
  "dependencies": {
    "es-errors": "^1.3.0",
    "object-inspect": "^1.13.3"
  },
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},

```

```

"node_modules/side-channel-map": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/side-channel-map/-/side-channel-map-1.0.1.tgz",
  "integrity": "sha512-VCjCNfgMsby3tTdo02nbjtM/ewra6jPHmpThenkTYh8pG9ucZ/1P8So4u4FGBek/BjpOVsDCMoLA/iuBKI
FXRA==",
  "dependencies": {
    "call-bound": "^1.0.2",
    "es-errors": "^1.3.0",
    "get-intrinsic": "^1.2.5",
    "object-inspect": "^1.13.3"
  },
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/side-channel-weakmap": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/side-channel-weakmap/-/side-channel-weakmap-1.0.2.tgz",
  "integrity": "sha512-WPS/HvHQTynHisLo9McqBHOJk2FkHO/tlpvldyrnem4acQp4hai3gythswg6p01oSoTl58rcpiFAjF2br2Ak
2A==",
  "dependencies": {
    "call-bound": "^1.0.2",
    "es-errors": "^1.3.0",
    "get-intrinsic": "^1.2.5",
    "object-inspect": "^1.13.3",
    "side-channel-map": "^1.0.1"
  },
  "engines": {
    "node": ">= 0.4"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/sift": {
  "version": "17.1.3",
  "resolved": "https://registry.npmjs.org/sift/-/sift-17.1.3.tgz",
  "integrity": "sha512-
Rtlj66/b0ICeFzYTuNvX/EF1igRbbnGSvEyT79McoZa/DeGhMyC5pWKOEsZKnPKqtSeovd5FL/bjHWC3
CIIvCQ=="
},
"node_modules/socket.io": {
  "version": "4.8.1",
  "resolved": "https://registry.npmjs.org/socket.io/-/socket.io-4.8.1.tgz",
  "integrity": "sha512-
oZ7iUCxph8WYRRHcjBEc9unw3adt5CmSNlppj/5Q4k2RIrhl8Z5yY2Xr4j9zj0+wzVZ0bxmYoGSzKJnRl6
A4yg==",
  "dependencies": {
    "accepts": "~1.3.4",
    "base64id": "~2.0.0",

```

```

    "cors": "~2.8.5",
    "debug": "~4.3.2",
    "engine.io": "~6.6.0",
    "socket.io-adapter": "~2.5.2",
    "socket.io-parser": "~4.2.4"
  },
  "engines": {
    "node": ">=10.2.0"
  }
},
"node_modules/socket.io-adapter": {
  "version": "2.5.5",
  "resolved": "https://registry.npmjs.org/socket.io-adapter/-/socket.io-adapter-2.5.5.tgz",
  "integrity": "sha512-eLDQas5dzPgOWCk9GuuJC2lBqItuhKI4uxGgo9aIV7MYbk2h9Q6uULEh8WBzTholI7l+qU9Ast9fVUmkqPP9wYg==",
  "dependencies": {
    "debug": "~4.3.4",
    "ws": "~8.17.1"
  }
},
"node_modules/socket.io-adapter/node_modules/debug": {
  "version": "4.3.7",
  "resolved": "https://registry.npmjs.org/debug/-/debug-4.3.7.tgz",
  "integrity": "sha512-Er2nc/H7RrMXZBFCEim6TCmMk02Z8vLC2Rbi1KEBggpo0fS6l0S1nnapwmlI3yW/+GOJap1Krg4w0Hg80oCqgQ==",
  "dependencies": {
    "ms": "^2.1.3"
  },
  "engines": {
    "node": ">=6.0"
  },
  "peerDependenciesMeta": {
    "supports-color": {
      "optional": true
    }
  }
},
"node_modules/socket.io-adapter/node_modules/ms": {
  "version": "2.1.3",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
  "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjqz5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA=="
},
"node_modules/socket.io-parser": {
  "version": "4.2.4",
  "resolved": "https://registry.npmjs.org/socket.io-parser/-/socket.io-parser-4.2.4.tgz",
  "integrity": "sha512-/GbIkmo8ioc+NIWIhwdecY0ge+qVBSMdgxGygevmdHj24bsfgtCmcUUcQ5ZzcylGFHsN3k4HB4Cgkl96KVnuew==",
  "dependencies": {
    "@socket.io/component-emitter": "~3.1.0",
    "debug": "~4.3.1"
  }
}

```

```

    },
    "engines": {
      "node": ">=10.0.0"
    }
  },
  "node_modules/socket.io-parser/node_modules/debug": {
    "version": "4.3.7",
    "resolved": "https://registry.npmjs.org/debug/-/debug-4.3.7.tgz",
    "integrity": "sha512-Er2nc/H7RrMXZBFCEim6TCmMk02Z8vLC2Rbi1KEBggpo0fS6l0S1nnapwmlI3yW/+GOJap1Krg4w0Hg80oCqgQ==",
    "dependencies": {
      "ms": "^2.1.3"
    },
    "engines": {
      "node": ">=6.0"
    },
    "peerDependenciesMeta": {
      "supports-color": {
        "optional": true
      }
    }
  },
  "node_modules/socket.io-parser/node_modules/ms": {
    "version": "2.1.3",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
    "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjqz5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
    },
    "node_modules/socket.io/node_modules/debug": {
      "version": "4.3.7",
      "resolved": "https://registry.npmjs.org/debug/-/debug-4.3.7.tgz",
      "integrity": "sha512-Er2nc/H7RrMXZBFCEim6TCmMk02Z8vLC2Rbi1KEBggpo0fS6l0S1nnapwmlI3yW/+GOJap1Krg4w0Hg80oCqgQ==",
      "dependencies": {
        "ms": "^2.1.3"
      },
      "engines": {
        "node": ">=6.0"
      },
      "peerDependenciesMeta": {
        "supports-color": {
          "optional": true
        }
      }
    },
    "node_modules/socket.io/node_modules/ms": {
      "version": "2.1.3",
      "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
      "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjqz5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTIA==",
      },

```

```

"node_modules/sparse-bitfield": {
  "version": "3.0.3",
  "resolved": "https://registry.npmjs.org/sparse-bitfield/-/sparse-bitfield-3.0.3.tgz",
  "integrity": "sha512-kvzhi7vqKtFkh0PZU+2D2Pillw2ymqJKUjUcyPMd9Y75Nv4nPbGJZXNhxsQdQab2BmlDct1YnfQCguEvHr7VsQ==",
  "dependencies": {
    "memory-pager": "^1.0.2"
  }
},
"node_modules/statuses": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",
  "integrity": "sha512-13d10E6bRPL4pZy2iXmP8GfIv0z0L55vwDYo1JcV8Z7S0P/nrxpd78GxvIdk42Kh1e01oLStU31j1PwL+g==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/streamsearch": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/streamsearch/-/streamsearch-1.1.0.tgz",
  "integrity": "sha512-Mcc5wHehp9aXz1ax6bZUyY5afg9u2rv5cqQI3mRrYkGC8rW2hM02jWuwjtL++LS5qinSyhj2QfLyNsuc+VsExg==",
  "engines": {
    "node": ">=10.0.0"
  }
},
"node_modules/string_decoder": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
  "integrity": "sha512-n/ShnvDi6FHbbVfviro+WojiFzv+s8MPMHBczVePfUpDJLwoLT0ht1l4YwBCbi8pJAveEEdnkHyPyTP/mzRfwg==",
  "dependencies": {
    "safe-buffer": "~5.1.0"
  }
},
"node_modules/string_decoder/node_modules/safe-buffer": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
  "integrity": "sha512-Gd2UZBJDkXlY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpScSmMJSOfF8LOIrt8ud/wPtojs4G6+g=="
},
"node_modules/toidentifier": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",
  "integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTIZUtEe+JG7s5YhEz608PIA HRA==",
  "engines": {
    "node": ">=0.6"
  }
}

```

```

    }
  },
  "node_modules/tr46": {
    "version": "5.0.0",
    "resolved": "https://registry.npmjs.org/tr46/-/tr46-5.0.0.tgz",
    "integrity": "sha512-tk2G5R2KRwBd+ZN0zaEXpmzdKyOYksXwywulIX95MBODjSzMIuQnQ3m8Jxgbhnl1LeVo7lqQKsYa1O3Htl7K5g==",
    "dependencies": {
      "punycode": "^2.3.1"
    },
    "engines": {
      "node": ">=18"
    }
  },
  "node_modules/type-is": {
    "version": "1.6.18",
    "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
    "integrity": "sha512-TkRr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+ma9BGm06XQBy8g==",
    "dependencies": {
      "media-typer": "0.3.0",
      "mime-types": "~2.1.24"
    },
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/typedarray": {
    "version": "0.0.6",
    "resolved": "https://registry.npmjs.org/typedarray/-/typedarray-0.0.6.tgz",
    "integrity": "sha512-/aCDEGatGvZ2BIk+HmLf4ifCJFwvKFNb9/JeZPMulfGFracn9QFcAf5GO8B/mweUjSoblS5In0cWhqpfs/5PQA==",
    },
  "node_modules/undici-types": {
    "version": "6.20.0",
    "resolved": "https://registry.npmjs.org/undici-types/-/undici-types-6.20.0.tgz",
    "integrity": "sha512-Ny6QQZ2Nju20vw1SRHe3d9jVu6gJ+4e3+MMpqu7pqE5HT6WsTSlce++GQmK5UXS8mzV8DSYHrQH+Xrf2jVcuKNg==",
    },
  "node_modules/unpipe": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
    "integrity": "sha512-pjy2bYhSsufwWIKwPc+l3cN7+wuJIK6uz0YdJEOlQDbl6jo/YIPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hztKQ==",
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/util-deprecate": {
    "version": "1.0.2",

```

```

    "resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",
    "integrity": "sha512-EPD5q1uXyFxJpCrLnCc1nHnq3gOa6DZBocAlIi2TaSCA7VCJ1UJDMagCzIkXNsUYfD1daK//LTEQ8xiI
brHtcw=="
  },
  "node_modules/utils-merge": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",
    "integrity": "sha512-npZDqzv20eKORC9l400GNGHnUpVWfHT32m3a1S10WjovwFuvJCF5Z4H4YkH6Xl5lWp4XzUuH9rXZ1gR
pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbbsHHoED/ztMtkv/VoYTYy
ShUn81hA==",
    "engines": {
      "node": ">= 0.4.0"
    }
  },
  "node_modules/vary": {
    "version": "1.1.2",
    "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
    "integrity": "sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOzmAKzYqYRAISoRhdQr
3eIZ/PXqg==",
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/webidl-conversions": {
    "version": "7.0.0",
    "resolved": "https://registry.npmjs.org/webidl-conversions/-/webidl-conversions-7.0.0.tgz",
    "integrity": "sha512-VwddBukDzu71offAQR975unBIGqfKZpM+8ZX6ySk8nYhVoo5CYaZyzt3YBvYtRtO+aoGlqxPg/B87NG
VZ/fu6g==",
    "engines": {
      "node": ">=12"
    }
  },
  "node_modules/whatwg-url": {
    "version": "14.1.0",
    "resolved": "https://registry.npmjs.org/whatwg-url/-/whatwg-url-14.1.0.tgz",
    "integrity": "sha512-jlf/foYIKywAt3x/XWKZ/3rz8OSJPiWktjmk891alJUEjiVxKX9LEO92qH3hv4aJ0mN3MWPvGMCy8jQi9
5xK4w==",
    "dependencies": {
      "tr46": "^5.0.0",
      "webidl-conversions": "^7.0.0"
    },
    "engines": {
      "node": ">=18"
    }
  },
  "node_modules/ws": {
    "version": "8.17.1",
    "resolved": "https://registry.npmjs.org/ws/-/ws-8.17.1.tgz",
    "integrity": "sha512-6XQFvXTkbfUOZOKKILFG1PDK2NDQs4azKQl26T0YS5CxxqWLgXajbPZ+h4gZekJyRqFU8pvnAbbs/
3TgRPY+GQ==",

```



```
"engines": {
  "node": ">=10.0.0"
},
"peerDependencies": {
  "bufferutil": "^4.0.1",
  "utf-8-validate": ">=5.0.2"
},
"peerDependenciesMeta": {
  "bufferutil": {
    "optional": true
  },
  "utf-8-validate": {
    "optional": true
  }
},
"node_modules/xtend": {
  "version": "4.0.2",
  "resolved": "https://registry.npmjs.org/xtend/-/xtend-4.0.2.tgz",
  "integrity": "sha512-LKYU1iAXJXUGAXn9URjiu+MWhyUXHsvfp7mcuYm9dSUKK0/CjtrUwFAxD82/mCWbtLsGjFIad0wIs
od4zrTAEQ==",
  "engines": {
    "node": ">=0.4"
  }
}
}
```

Testing:

| Test Case ID | Test Case | Expected Result | Status |
|--------------|--|---|--------|
| TC_01 | Login with valid credentials | User is redirected to dashboard | Passed |
| TC_02 | Login with invalid credentials | "Access Denied" message is shown | Passed |
| TC_03 | Add a new employee | Employee is successfully added | Passed |
| TC_04 | Edit existing employee details | Employee data is updated | Passed |
| TC_05 | Delete an employee | Employee record is removed | Passed |
| TC_06 | Submit a leave application | Leave is saved and marked as "Pending" | Passed |
| TC_07 | Approve a leave request | Status changes to "Approved" | Passed |
| TC_08 | Reject a leave request | Status changes to "Rejected" | Passed |
| TC_09 | View all leave requests | List of leave applications is displayed | Passed |
| TC_10 | Unauthorized user tries to access admin features | "Access Denied" message shown | Passed |

Sidebar code:

```
import React, { useState } from "react";
import "./sidebar.css";

import { FaUsers } from "react-icons/fa";
import { RiBriefcase4Fill } from "react-icons/ri";
import { IoGridOutline } from "react-icons/io5";
import { MdOutlineProductionQuantityLimits } from "react-icons/md";
import { MdHistoryEdu } from "react-icons/md";
import { MdOutlineCategory } from "react-icons/md";
import { Link } from "react-router-dom";

const SideBar = () => {
  let role = "";

  const [dropdowns, setDropdowns] = useState({
    employees: false,
    categorys: false,
    product: false,
    drivers: false,
    proposals: false,
    leave: false,
  });

  const toggleDropdown = (name) => {
    setDropdowns({ ...dropdowns, [name]: !dropdowns[name] });
  };

  role = sessionStorage.getItem("role");
  return (
    <aside id="sidebar" className="sidebar">
      <ul className="sidebar-nav" id="sidebar-nav">
        { /* <li className='nav-item'>
          <Link className='nav-link' to='/dashboard'>
            <IoGridOutline size={20} />
            <span className='nav-heading collapsed'>Dashboard</span>
          </Link>
        </li> */ }
        {role === "admin" && (
          <li className="nav-item">
            <Link className="nav-link" to="/admindashboard">
              <IoGridOutline size={20} />
              <span className="nav-heading collapsed">Admin Dashboard</span>
            </Link>
          </li>
        )}
        <li className="nav-item">
          <div
            className="nav-link"
            onClick={() => toggleDropdown("categorys")}
          >
            <MdOutlineCategory size={20} />
            <span className="nav-heading collapsed">Employee</span>
          </div>
        </li>
      </ul>
    </aside>
  );
}
```

```

</div>
{dropdowns.categorys && (
  <ul className="nav-content">
    <li className="ps-3">
      <Link to="/add-employee" className="nav-link">
        <i className="bi bi-circle"></i>
        <span className="nav-heading collapsed">
          Add Employee
        </span>
      </Link>
    </li>
    <li className="ps-3">
      <Link to="/employee-list" className="nav-link">
        <i className="bi bi-circle"></i>
        <span className="nav-heading collapsed">
          Employee list
        </span>
      </Link>
    </li>
  </ul>
)}
</li>
<li className="nav-item">
  <Link className="nav-link" to="/employees-leave">
    <IoGridOutline size={20} />
    <span className="nav-heading collapsed">Employee Leaves</span>
  </Link>
</li>

<li className="nav-item">
  <Link className="nav-link" to="/query-list">
    <MdHistoryEdu size={20} />
    <span className="nav-heading collapsed">All Querys</span>
  </Link>
</li>
</>
)}

{role === "employee" && (
  <
    <li className="nav-item">
      <Link className="nav-link" to="/franchisedashboard">
        <IoGridOutline size={20} />
        <span className="nav-heading collapsed">
          Franchise Dashboard
        </span>
      </Link>
    </li>
    <li className="nav-item">
      <Link className="nav-link" to="/send-query">
        <IoGridOutline size={20} />
        <span className="nav-heading collapsed">Raise Query</span>
      </Link>
    </li>
    <li className="nav-item">

```

```

    <Link className="nav-link" to="/your-query">
      <IoGridOutline size={20} />
      <span className="nav-heading collapsed">Your query</span>
    </Link>
  </li>

  <li className="nav-item">
    <div className="nav-link" onClick={() => toggleDropdown("leave")}>
      <RiBriefcase4Fill size={20} />
      <span className="nav-heading collapsed">Leaves</span>
    </div>
    {dropdowns.leave && (
      <ul className="nav-content">
        <li className="ps-3">
          <Link to="/apply-leave" className="nav-link">
            <i className="bi bi-circle"></i>
            <span className="nav-heading collapsed">Apply Leave</span>
          </Link>
        </li>
        <li className="ps-3">
          <Link to="/leave-history" className="nav-link">
            <i className="bi bi-circle"></i>
            <span className="nav-heading collapsed">
              Leave history
            </span>
          </Link>
        </li>
      </ul>
    )}
  </li>
</>
)}
</ul>
</aside>
);
};

export default SideBar;

```

Main Routes:

```

import React, { useEffect, useState } from "react";
import { Routes, Route, useLocation } from "react-router-dom";

import Header from "../Header.jsx";
import SideBar from "../SideBar.jsx";

import "../main.css";
import AdminDashboard from "../Pages/Admin/AdminDashboard.jsx";
import PageTitle from "../PageTitle.jsx";
import FranchiseDashboard from "../Pages/Employee/FranchiseDashboard.jsx";
import Protected from "../Pages/Protected.jsx";

```

```

import Addquery from "../Pages/Employee/Addquery.jsx";
import Addemployee from "../Pages/Admin/Addemployee.jsx";
import Employeeelist from "../Pages/Admin/Employeeelist.jsx";
import Viewquery from "../Pages/Admin/Viewquery.jsx";
import Viewourquery from "../Pages/Employee/Viewourquery.jsx";
import Leavelist from "../Pages/Admin/Leavelist.jsx";
import Applyleave from "../Pages/Employee/Applyleave.jsx";
import Leavehistory from "../Pages/Employee/Leavehistory.jsx";

const Allmain = () => {
  const location = useLocation();
  const [pageTitle, setPageTitle] = useState("");

  useEffect(() => {
    // // Map routes to page titles
    const routeToTitle = {
      "/dashboard": "Dashboard",
      "/franchisedashboard": "Franchise dashboard",
      "/add-employee": "Add Employee",
      "/employee-list": "Employee List",
      "/send-query": "Send Query",
      "/query-list": "All queries",
      "/your-query": "Your Query",
      "/employees-leave": "Employee Leaves",
      "/apply-leave": "Apply Leave",
      "/leave-history": "Leave History"
    };

    const title = routeToTitle[location.pathname];
    if (title) {
      setPageTitle(title);
    } else {
      setPageTitle("");
    }
  }, [location.pathname]);

  return (
    <>
      <Header />
      <SideBar />
      <main
        id="main"
        className="main"
        style={{ backgroundColor: "#D6E9FC", height: "100vh" }}
      >
        { /* <PageTitle page={pageTitle} /> */ }
        <Routes>
          <Route path="/admindashboard" element={ <AdminDashboard /> } />
          <Route path="/franchisedashboard" element={ <FranchiseDashboard /> } />
          <Route path="/add-employee" element={ <Addemployee /> } />
          <Route path="/employee-list" element={ <Employeeelist /> } />
          <Route path="/send-query" element={ <Addquery /> } />
          <Route path="/query-list" element={ <Viewquery /> } />
          <Route path="/your-query" element={ <Viewourquery /> } />
          <Route path="/employees-leave" element={ <Leavelist /> } />
          <Route path="/apply-leave" element={ <Applyleave /> } />
        </Routes>
      </main>
    </>
  );
}

```

```

    <Route path="/leave-history" element={<Leavehistory />} />

    { /* user Billing routes */ }
  </Routes>
</main>
</>
);
};

export default Allmain;

```

Login Page:

```

import React, { useState } from 'react'
import { useNavigate } from 'react-router-dom';
import "bootstrap/dist/css/bootstrap.css";
import './login.css';
import Swal from 'sweetalert2';
import axios from 'axios';
const Login = () => {
  const [id, setId] = useState(""); // Single state for adminId or employeeId
  const [password, setPassword] = useState("");
  const navigate = useNavigate();

  // Handle login submission
  const handleLogin = async (e) => {
    e.preventDefault();

    // Ensure the ID and password are provided
    if (!id || !password) {
      Swal.fire("Error", "Please enter ID and password", "error");
      return;
    }

    try {
      // We will conditionally send either adminId or employeeId
      const body = {};
      if (id.includes("admin")) {
        body.adminId = id; // Set adminId if ID contains 'admin' (or based on your logic)
      } else {
        body.employeeId = id; // Otherwise, assume it's an employeeId
      }

      body.password = password; // Always send password

      const response = await axios.post(
        "http://localhost:5000/hr/admin/login",
        body
      );
      // console.log("response of login api", response);
    }
  }
}

```

```

if (response.status === 200) {
  Swal.fire({
    icon: "success",
    title: "Login successful!",
    text: "Please enter the OTP sent to your email.",
  });

  // Save the token and role in session storage
  sessionStorage.setItem("token", response.data.token);
  sessionStorage.setItem("role", response.data.role);

  // Redirect based on role
  const role = response.data.role;
  const token = sessionStorage.getItem("token");

  if (!token || !role) {
    navigate("/");
  } else if (role === "admin") {
    navigate("/admindashboard"); // Redirect to admin dashboard
  } else if (role === "employee") {
    navigate("/franchisedashboard"); // Redirect to franchise dashboard
  }
}
} catch (error) {
  Swal.fire(
    "Error",
    error.response?.data?.message || "Login failed",
    "error"
  );
}
};
return (
  <
    <section className="vh-100" style={{ backgroundColor: "#fff" }}>
      <div className="container py-5 h-100">
        <div className="row d-flex align-items-center justify-content-center h-100">
          <div className="col-md-8 col-lg-7 col-xl-6">
            
          </div>
          <div className="col-md-7 col-lg-5 col-xl-5 offset-xl-1">
            <form onSubmit={handleLogin}>
              {/* ID Field for either Admin or Employee */}
              <div className="form-outline mb-4">
                <input
                  type="text"
                  className="form-control form-control-lg"
                  placeholder="Admin ID or Employee ID"
                  value={id}
                  onChange={(e) => setId(e.target.value)}
                  required
                />
              </div>
            </form>
          </div>
        </div>
      </div>
    </section>
  )
)

```



```

    </div>

    { /* Password Field */}
    <div className="form-outline mb-4">
      <input
        type="password"
        className="form-control form-control-lg"
        placeholder="Password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        required
      />
    </div>

    { /* Submit Button */}
    <button
      type="submit"
      className="btn btn-primary btn-lg btn-block"
    >
      Sign in
    </button>
  </form>
</div>
</div>
</div>
</section>
</>
)
}

export default Login

```

Admin Dashboard:

```
import React, { useEffect, useState } from "react";
import { FaUsers } from "react-icons/fa6";
import { MdOutlineQueryStats } from "react-icons/md";
import { FaArrowUp } from "react-icons/fa";
import { FaRegSmile } from "react-icons/fa";
import { TbTruckDelivery } from "react-icons/tb";
import rulebook from "../images/Dream-rule-book.pdf";
import {
  BarChart,
  Bar,
  XAxis,
  YAxis,
  CartesianGrid,
  Tooltip,
  Legend,
  ResponsiveContainer,
  PieChart,
  Pie,
  Cell,
} from "recharts";
import "../admin.css";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import PageTitle from "../components/PageTitle";

const AdminDashboard = () => {
  const token = sessionStorage.getItem("token");
  const navigate = useNavigate();

  const [employeeData, setEmployeeData] = useState(0);
  const [unsolvedData, setUnsolvedData] = useState(0);
  const [solvedData, setSolvedData] = useState(0);
  const [employeeChartData, setEmployeeChartData] = useState([]);
  const [queryChartData, setQueryChartData] = useState([]);

  useEffect(() => {
    TotalEmployeeData();
    TotalUnsolvedQuery();
    TotalSolvedQuery();
  }, []);

  const TotalEmployeeData = async () => {
    try {
      const url = `http://localhost:5000/hr/admin/emplist`;
      const headers = {
        "Content-Type": "multipart/form-data",
        Accept: "application/json",
        Authorization: `Bearer ${token}`,
      };
      const response = await axios.get(url, { headers: headers });
      setEmployeeData(response.data.data.length);
    }
  }
}
```

```

const employeeData = response.data.data.map((emp) => ({
  name: emp.name,
  createdAt: new Date(emp.createdAt).toLocaleDateString(),
}));
setEmployeeChartData(employeeData);
} catch (error) {
  console.log(error);
}
};

const Totalunsolvequery = async () => {
  try {
    const url = `http://localhost:5000/hr/admin/getallquery`;
    const headers = {
      "Content-Type": "multipart/form-data",
      Accept: "application/json",
      Authorization: `Bearer ${token}`,
    };
    const response = await axios.get(url, { headers: headers });
    setunsolvedata(response.data.data.length);

    const queryData = response.data.data.reduce((acc, query) => {
      const date = new Date(query.createdAt).toLocaleDateString();
      acc[date] = (acc[date] || 0) + 1;
      return acc;
    }, {});
    const pieChartData = Object.keys(queryData).map((date) => ({
      name: date,
      value: queryData[date],
    }));
    setQueryChartData(pieChartData);
  } catch (error) {
    console.log(error);
  }
};

const Totalsolvequery = async () => {
  try {
    const url = `http://localhost:5000/hr/admin/solvequery`;
    const headers = {
      "Content-Type": "multipart/form-data",
      Accept: "application/json",
    };
    const response = await axios.get(url, { headers: headers });
    setsolvedata(response.data.data.length);
  } catch (error) {
    console.log(error);
  }
};

const viewPDF = () => {
  window.open(rulebool, "_blank");
};

const handleCardClick = (move) => {

```

```

if (typeof move === "string") {
  navigate(move); // Navigate to the specified route
} else if (typeof move === "function") {
  move(); // Execute the function (e.g., open PDF)
}
};

```

```

const cardData = [
  {
    title: "Total Employee",
    value: `${employeeData}`,
    percentage: "+30%",
    description: "Since last month",
    icon: <FaUsers />,
    iconBg: "#3c96ef",
    move: "/employee-list", // Route for navigation
    percentageBg: "bg-light-green",
  },
  {
    title: "Total query",
    value: `${unsolvedData}`,
    percentage: "+23%",
    description: "Since last week",
    icon: <MdOutlineQueryStats />,
    iconBg: "#51ab55",
    move: "/query-list", // Route for navigation
    percentageBg: "bg-light-green",
  },
  {
    title: "Total Solve query",
    value: `${solvedData}`,
    percentage: "-10%",
    description: "Since last month",
    icon: <FaRegSmile />,
    iconBg: "#de2768",
    move: "/query-list", // Route for navigation
    percentageBg: "bg-light-red",
  },
  {
    title: "Rule Book",
    value: "",
    icon: <TbTruckDelivery />,
    iconBg: "#272729",
    move: viewPDF, // Function to open PDF
  },
];

```

```

const COLORS = ["#FF5733", "#33FF57", "#5733FF", "#FFC300", "#FF33F6"];

```

```

return (
  <>
    <PageTitle page={"admin dashboard"} />
    <div className="container py-4" style={{ backgroundColor: "#D6E9FC" }}>
      <div className="row">
        {cardData.map((card, index) => (

```

```

<div className="col-md-3 mb-4" key={index}>
  <div
    className="card position-relative"
    style={{
      boxShadow:
        "rgba(0, 0, 0, 0.1) 0rem 0.25rem 0.375rem -0.0625rem, rgba(0, 0, 0, 0.06) 0rem 0.125rem
0.25rem -0.0625rem",
      backgroundColor: "#ffffff",
      borderRadius: "10px",
      border: "none",
    }}
    onClick={() => handleCardClick(card.move)} // Handle card click
  >
    <div
      className={`icon-wrapper position-absolute`}
      style={{
        backgroundColor: card.iconBg,
        width: "65px",
        height: "65px",
        borderRadius: "20%",
        display: "flex",
        alignItems: "center",
        justifyContent: "center",
        top: "-20px",
        right: "15px",
        fontSize: "35px",
        color: "white",
        boxShadow:
          "rgba(0, 0, 0, 0.14) 0rem 0.25rem 1.25rem 0rem, rgba(64, 64, 64, 0.4) 0rem 0.4375rem
0.625rem -0.3125rem",
      }}
    >
      {card.icon}
    </div>
    <div className="card-body">
      <h6 className="heading">{card.title}</h6>
      <h4 className="font-weight-bold title">
        {card.value} {" "}
        <span className="h6 text-success">
          <FaArrowUp /> {" "}
        </span>
      </h4>
      <p className="break"></p>
    </div>
  </div>
</div>
))}
</div>

{/* Graphs Section */}
<div className="row mt-4">
  <div className="col-md-6">
    <div className="card">
      <div className="card-body">
        <h5 className="card-title">Employee Overview</h5>

```

```

<ResponsiveContainer width="100%" height={300}>
  <BarChart data={employeeChartData}>
    <CartesianGrid strokeDasharray="3 3" />
    <XAxis dataKey="createdAt" />
    <YAxis />
    <Tooltip
      content={({ payload }) => {
        if (payload && payload[0]) {
          return (
            <div
              className="custom-tooltip"
              style={{ color: "red" }}
            >
              <p>`Name: ${payload[0].payload.name}`</p>
              <p>`Created At: ${payload[0].payload.createdAt}`</p>
            </div>
          );
        }
        return null;
      }}
    />
    <Legend />
    <Bar dataKey="createdAt" fill="#FF5733" />
  </BarChart>
</ResponsiveContainer>
</div>
</div>
</div>

<div className="col-md-6">
  <div className="card">
    <div className="card-body">
      <h5 className="card-title">Query Distribution</h5>
      <ResponsiveContainer width="100%" height={300}>
        <PieChart>
          <Pie
            data={queryChartData}
            cx="50%"
            cy="50%"
            labelLine={false}
            outerRadius={80}
            fill="#8884d8"
            dataKey="value"
            nameKey="name"
          >
            {queryChartData.map((entry, index) => (
              <Cell
                key={`cell-${index}`}
                fill={COLORS[index % COLORS.length]}
              />
            ))}
          </Pie>
          <Tooltip
            content={({ payload }) => {
              if (payload && payload[0]) {

```

```
        return (  
          <div className="custom-tooltip">  
            <p>{'Date: ${payload[0].payload.name}'}</p>  
            <p>{'Queries: ${payload[0].value}'}</p>  
          </div>  
        );  
      }  
      return null;  
    }  
  }  
  />  
  <Legend />  
  </PieChart>  
  </ResponsiveContainer>  
  </div>  
  </div>  
  </div>  
  </div>  
  </div>  
  </>  
);  
};
```

```
export default AdminDashboard;
```

Employee Dashboard:

```
import React, { useEffect, useState } from "react";
import "../franchise.css";
import axios from "axios";
import { jwtDecode } from "jwt-decode";
import PageTitle from "../../components/PageTitle";
import { Link } from "react-router-dom";
import { FaArrowUp } from "react-icons/fa";
import { MdProductionQuantityLimits } from "react-icons/md";
import rulebool from "../../images/Dream-rule-book.pdf";
import { TbTruckDelivery } from "react-icons/tb";

const FranchiseDashboard = () => {
  const token = sessionStorage.getItem("token");
  const decodedData = jwtDecode(token);
  const employeeId = decodedData.id;

  const [queryData, setQueryData] = useState(0);

  useEffect(() => {
    Getyourquery();
  }, []);

  const Getyourquery = async () => {
    try {
      const url = `http://localhost:5000/hr/employee/getquery/${employeeId}`;
      const headers = {
        "Content-Type": "application/json",
        Accept: "application/json",
        Authorization: `Bearer ${token}`,
      };

      const response = await axios.get(url, { headers: headers });
      // console.log("Response of query:", response);
      if (response.data && response.data.data) {
        setQueryData(response.data.data.length);
      }
    } catch (error) {
      console.log(error);
    }
  };

  const viewPDF = () => {
    window.open(rulebool, "_blank");
  };

  const cardData = [
    {
      title: "Total Query",
      value: queryData,
      percentage: "+30%",
      description: "Since last month",
      icon: <MdProductionQuantityLimits />,
      iconBg: "#3c96ef",
    },
  ],
```



```
    >
    <h6 className="heading">{card.title}</h6>
    <h4 className="font-weight-bold title">
      {card.value} {" "}
      <span className="h6 text-success">
        <FaArrowUp /> {" "}
      </span>
    </h4>
    <p className="break"></p>
  </Link>
</div>
</div>
</div>
  )}
</div>
</div>
</>
);
};
```

```
export default FranchiseDashboard;
```

Thank You