MediCare Solutions-A Complete Hosptial Management Portal

Tech to be used :

1. MERN stack
2. FRONT END,BACK END,TWO DASHBOARDS(admin and users)
3. Deploy on vercel/netlify

Requirements:

1. User Login/Signup
2. Admin Login/add new admin
3. All signup are jwt authorised
4. Admin Dashboard and Users Dashboard
5. Patient able to book OPD slots
6. Review the bookings and mark them as Booked/Complete/Incomplete
7. Send Query to hospital
8. …
9. …

Implementation :

Packages used:

· **bcrypt**: Used for hashing passwords securely to protect them in storage.

· **cloudinary**: A cloud-based service used for managing and delivering images and videos, often used to upload and store media files.

· **cookie-parser**: Middleware for parsing cookies attached to client requests, useful for managing session data.

· **cors**: Middleware to enable Cross-Origin Resource Sharing, allowing resources to be requested from another domain.

· **dotenv**: Loads environment variables from a .env file into process.env, useful for managing configuration settings.

· **express**: A minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

· **mongoose**: An ODM (Object Data Modeling) library for MongoDB and Node.js, providing a schema-based solution to model data.

· **express-fileupload**: Middleware that allows easy file uploading in Express applications.

· **jsonwebtoken**: Used for creating and verifying JSON Web Tokens (JWT), commonly used for authentication.

· **validator**: A library of string validators and sanitizers, used to validate and sanitize inputs in web applications.

ABOUT MONGODB ATLAS:

Data base name : MERN\_MediCare\_Solutions

Passsword amityadav1100

Name yadavamit5690

mongodb+srv://yadavamit5690:amityadav1100@cluster0.tbhmx.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0

* **Managed Database Service**: MongoDB Atlas is a fully managed cloud database service that handles the complexities of deploying, managing, and scaling MongoDB across cloud providers like AWS, Google Cloud, and Azure.
* **Multi-Region and Multi-Cloud**: It offers the ability to deploy in multiple regions or across different cloud providers to enhance availability and reduce latency.
* **Automated Backups and Security**: Atlas provides automated backups, monitoring, and advanced security features like encryption and access control.

**Cross-Origin Resource Sharing (CORS) :**refers to the situations when a frontend running in a browser has JavaScript code that communicates with a backend, and the backend is in a different "origin" than the frontend.

- CORS (Cross-Origin Resource Sharing) allows servers to specify which origins are permitted to access their resources via new HTTP headers.

- It controls access by setting rules in headers for different origins trying to read data from a browser.

- For HTTP methods that may alter server data (like POST, PUT, DELETE, etc.), CORS requires a "preflight" request using the HTTP OPTIONS method.

- The preflight request checks with the server if the actual request is allowed, and only proceeds if the server approves.

- Servers can specify if credentials (Cookies, HTTP Authentication) should be included in requests.

- CORS failures result in errors, but the specifics are hidden from JavaScript for security reasons.

To understand CORS errors, you must check the browser’s console for detailed information.

Async , req.body, req,res

1.03.08