

CODING:

```
<!DOCTYPE html>

<html>

<head>

  <title>Add Contact</title>

</head>

<body>

  <h2>Add Contact</h2>

  <form id="contactForm">

    Name: <input type="text" id="name"><br>

    Phone: <input type="text" id="phone"><br>

    Email: <input type="text" id="email"><br>

    Address: <input type="text" id="address"><br>

    <button type="submit">Submit</button>

  </form>

  <script>

    document.getElementById("contactForm").addEventListener("submit", async function(e) {

      e.preventDefault();

      let response = await fetch("http://localhost:5000/api/contacts", {

        method: "POST",

        headers: { "Content-Type": "application/json" },

        body: JSON.stringify({

          name: document.getElementById("name").value,

          phone: document.getElementById("phone").value,

          email: document.getElementById("email").value,

          address: document.getElementById("address").value

        })

      })
```

```
});  
let result = await response.json();  
alert(JSON.stringify(result));  
});  
</script>  
</body>  
</html>
```

frontend/view.html :

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Contacts List</title>  
  <style>  
    body { font-family: Arial, sans-serif; margin: 20px; }  
    h2 { color: #333; }  
    table { border-collapse: collapse; width: 100%; margin-top: 15px; }  
    th, td { border: 1px solid #ccc; padding: 8px; text-align: left; }  
    th { background: #f4f4f4; }  
    tr:nth-child(even) { background: #f9f9f9; }  
  </style>  
</head>  
<body>  
  <h2>Saved Contacts</h2>  
  <table id="contactsTable">  
    <thead>  
      <tr>  
        <th>Name</th>
```

```
<th>Phone</th>
<th>Email</th>
<th>Address</th>
</tr>
</thead>
<tbody></tbody>
</table>
<script>
  async function fetchContacts() {
    let response = await fetch("http://localhost:5000/api/contacts");
    let contacts = await response.json();
    let tableBody = document.querySelector("#contactsTable tbody");
    tableBody.innerHTML = "";
    contacts.forEach(contact => {
      let row = `<tr>
        <td>${contact.name}</td>
        <td>${contact.phone}</td>
        <td>${contact.email}</td>
        <td>${contact.address}</td>
      </tr>`;
      tableBody.innerHTML += row;
    });
  }
  fetchContacts();
</script>
</body>
</html>
```

src/ config/db.js :

```
const mongoose = require("mongoose");
const connectDB = async () => {
  try {
    await mongoose.connect("mongodb://127.0.0.1:27017/contactDB");
    console.log(" MongoDB Connected");
  } catch (error) {
    console.error(" MongoDB Connection Failed:", error.message);
    process.exit(1);
  }
};
module.exports = connectDB;
```

src/ controllers/contactControllers.js:

```
const Contact = require("../models/Contact");
const { contactValidation } = require("../services/contactService");
exports.createContact = async (req, res) => {
  const { error } = contactValidation(req.body);
  if (error) return res.status(400).json({ error: error.details[0].message });
  try {
    const exists = await Contact.findOne({
      $or: [{ phone: req.body.phone }, { email: req.body.email }],
    });
    if (exists) return res.status(400).json({ error: "Phone/Email already exists" });
    const contact = new Contact(req.body);
    await contact.save();
    res.json({ status: "success", message: "Contact saved successfully", contact });
  } catch (err) {
```

```
    res.status(500).json({ error: err.message });
  }
};

exports.getContacts = async (req, res) => {
  const contacts = await Contact.find();
  res.json(contacts);
};

exports.getContactById = async (req, res) => {
  try {
    const contact = await Contact.findById(req.params.id);
    if (!contact) return res.status(404).json({ error: "Contact not found" });
    res.json(contact);
  } catch (err) {
    res.status(400).json({ error: "Invalid ID" });
  }
};

exports.updateContact = async (req, res) => {
  const { error } = contactValidation(req.body);
  if (error) return res.status(400).json({ error: error.details[0].message });
  try {
    const contact = await Contact.findByIdAndUpdate(req.params.id, req.body, { new: true });
    if (!contact) return res.status(404).json({ error: "Contact not found" });
    res.json({ status: "success", message: "Contact updated", contact });
  } catch (err) {
    res.status(400).json({ error: err.message });
  }
};
```

```
exports.deleteContact = async (req, res) => {  
  try {  
    const contact = await Contact.findByIdAndDelete(req.params.id);  
    if (!contact) return res.status(404).json({ error: "Contact not found" });  
    res.json({ status: "success", message: "Contact deleted" });  
  } catch (err) {  
    res.status(400).json({ error: "Invalid ID" });  
  }  
};
```

src/models/ Contact.js:

```
const mongoose = require("mongoose");  
const contactSchema = new mongoose.Schema({  
  name: { type: String, required: true },  
  phone: { type: String, required: true, unique: true },  
  email: { type: String, required: true, unique: true },  
  address: { type: String, required: true },  
});  
module.exports = mongoose.model("Contact", contactSchema);
```

src/routes/contactRoutes.js:

```
const express = require("express");  
const router = express.Router();  
const contactController = require("../controllers/contactController");  
router.post("/contacts", contactController.createContact);  
router.get("/contacts", contactController.getContacts);  
router.get("/contacts/:id", contactController.getContactById);  
router.put("/contacts/:id", contactController.updateContact);  
router.delete("/contacts/:id", contactController.deleteContact);
```

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

src/services/contactServices.js:

```
const Joi = require("joi");

const contactValidation = (data) => {

  const schema = Joi.object({

    name: Joi.string().pattern(/^[A-Za-z ]+$/).required(),

    phone: Joi.string().pattern(/^[0-9]{10,}$/).required(),

    email: Joi.string().email().required(),

    address: Joi.string().required(),

  });

  return schema.validate(data);

};

module.exports = { contactValidation };
```

src/server.js/:

```
const express = require("express");
const path = require("path");
const connectDB = require("./src/config/db");
const contactRoutes = require("./src/routes/contactRoutes");
const app = express();
connectDB();
app.use(express.json());
//  Serve frontend files
app.use("/frontend", express.static(path.join(__dirname, "frontend")));
// API routes
app.use("/api", contactRoutes);
const PORT = 5000;
app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));
```

