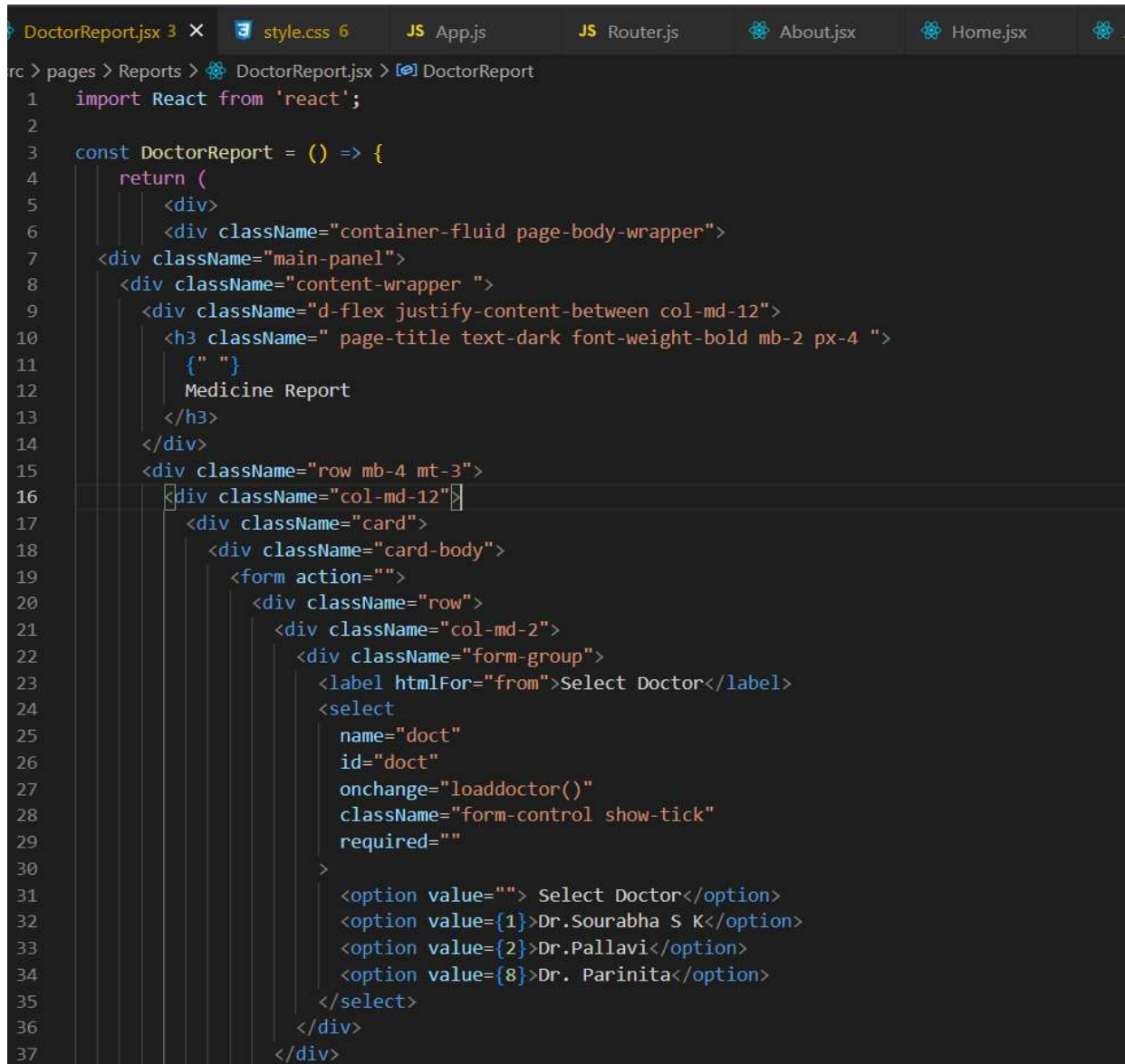


Clinical Management for Ayurveda Doctors

Front-End Screen Shots Snippets:



```
1  import React from 'react';
2
3  const DoctorReport = () => {
4    return (
5      <div>
6        <div className="container-fluid page-body-wrapper">
7          <div className="main-panel">
8            <div className="content-wrapper ">
9              <div className="d-flex justify-content-between col-md-12">
10               <h3 className=" page-title text-dark font-weight-bold mb-2 px-4 ">
11                 {" "}
12                 Medicine Report
13               </h3>
14             </div>
15             <div className="row mb-4 mt-3">
16               <div className="col-md-12">
17                 <div className="card">
18                   <div className="card-body">
19                     <form action="">
20                       <div className="row">
21                         <div className="col-md-2">
22                           <div className="form-group">
23                             <label htmlFor="from">Select Doctor</label>
24                             <select
25                               name="doct"
26                               id="doct"
27                               onChange="loaddoctor()"
28                               className="form-control show-tick"
29                               required=""
30                             >
31                               <option value=""> Select Doctor</option>
32                               <option value={1}>Dr.Sourabha S K</option>
33                               <option value={2}>Dr.Pallavi</option>
34                               <option value={8}>Dr. Parinita</option>
35                             </select>
36                           </div>
37                         </div>

```

Fig 1- Medicine Report

```
6 JS App.js JS Router.js About.jsx Home.jsx AddDoctor.jsx 4 index.css
c > pages > Appointments > AddAppointment.jsx > ...
1 import React from 'react'
2
3 const AddAppointment = () => {
4   return (
5     <div>
6       <div className="container-fluid page-body-wrapper">
7         <div className="main-panel">
8           <div className="content-wrapper">
9             <h3 className="text-dark font-weight-bold mb-2 px-4">
10               Patient Registration
11             </h3>
12             <div className="row p-4">
13               <div className="col-md-12">
14                 <div className="card">
15                   <div className="card-body">
16                     <form
17                       action="Process/ProcessAppointment.php"
18                       className="forms-sample"
19                       method="post"
20                     >
21                       <div className="row">
22                         <div className="col-md-4">
23                           <div className="form-group">
24                             <label htmlFor="Patientname">
25                               Patient Name <small className="text-danger">*</small>
26                             </label>
27                             <input
28                               type="text"
29                               className="form-control"
30                               name="patient_name"
31                               id="Patientname"
32                               placeholder="Patient name"
33                               required=""
34                             />
35                           </div>
36                         </div>
37                       <div className="col-md-4">
```

Fig 2 – Patient Registration Details

```
jsx Home.jsx AddDoctor.jsx 4 index.css LabTestGroup.jsx 9+ AddAppointment.jsx 1
c > pages > Letters > ReferenceLetter.jsx > ReferenceLetter
1 import React from 'react'
2
3 const ReferenceLetter = () => {
4   return (
5     <div>
6       <div className="container-fluid page-body-wrapper">
7         <div className="main-panel">
8           <div className="content-wrapper">
9             <h3 className="page-title text-dark font-weight-bold mb-2 px-4">
10               Referring Doctor
11             </h3>
12             <div className="row p-4">
13               <div className="col-md-12">
14                 <div className="card">
15                   <div className="card-body">
16                     <h4>To,</h4>
17                     <form action="Process/processLetter.php" method="POST">
18                       <div className="d-flex align-items-center mt-4">
19                         <h4 className="mr-3">Dr. </h4>
20                         <input
21                           type="text"
22                           className="form-control mw-100 w-25 ml-5"
23                           placeholder="Enter Doctor Name"
24                           name="drname"
25                           id="drname"
26                         />
27                       </div>
28                       <div className="row mt-5">
29                         <div className="col-md-12 txt text-center">
30                           <h5>Many Thanks for the referral.</h5>
31                         </div>
32                         <div className="col-md-12 mt-4">
33                           <h5
34                             className="d-flex align-items-center flex-wrap"
35                             style={{
36                               textAlign: "justify",
37                               textJustify: "inter-word"
38                             }}
39                         >
```

Fig 3 – Referring Doctor

```
rc > pages > Pharmacy > MedicalBilling.jsx > ...
1  import React from 'react'
2
3  const MedicalBilling = () => {
4    return (
5      <div>
6        <div className="container-fluid page-body-wrapper">
7          <div className="main-panel">
8            <div className="content-wrapper">
9              <h3 className="page-title text-dark font-weight-bold mb-2 px-4">
10                Pharmacy Bill
11              </h3>
12              <div className="row p-4">
13                <div className="col-md-12">
14                  <div className="card">
15                    <div className="card-body">
16                      <div className="table-responsive" style={{ width: "100%" }}>
17                        <div
18                          id="example_wrapper"
19                          className="dataTables_wrapper dt-bootstrap4 no-footer"
20                        >
21                          <div className="row">
22                            <div className="col-sm-12 col-md-6">
23                              <div className="dataTables_length" id="example_length">
24                                <label>
25                                  Show { " " }
26                                  <select
27                                    name="example_length"
28                                    aria-controls="example"
29                                    className="custom-select custom-select-sm form-control form-control-sm"
30                                  >
31                                    <option value={10}>10</option>
32                                    <option value={25}>25</option>
33                                    <option value={50}>50</option>
34                                    <option value={100}>100</option>
35                                  </select> { " " }
36                                  entries
37                                </label>
38                              </div>
```

Fig 4 – Pharmacy Bill

Backend Code Snippets:



```
JS server.js M X [Preview] README.md
JS server.js > ...
1  const express = require("express");
2  const app = express();
3  const cors = require("cors");
4  const dotenv = require("dotenv");
5  dotenv.config();
6
7  const db = require("../app/models");
8
9  const { notFound, errorHandler } = require("../app/middlewares/errorMiddleware")
10
11
12
13  db.sequelize.sync({});
14
15  // support cors
16  app.use(cors());
17
18
19  // parse application/json
20  app.use(express.json({limit: '50mb'}));
21
22  // parse x-www-form-urlencoded
23  app.use(express.urlencoded({ extended: true }));
24
25  // simple route
26  app.get("/", (req, res) => {
27    |   return res.status(200).json({ message: "Namaste duniya waalon!" });
28  | });
29
30  // Middlewares
31  require("../app/routes/data.routes")(app);
32  require("../app/routes/user.routes")(app);
33  require("../app/routes/employee.routes")(app);
34  require("../app/routes/lab_test.routes")(app);
35
36
37  app.use(notFound);
38  app.use(errorHandler);
39
40  // set port, listen for requests
41  const PORT = 8000;
42
43  app.listen(PORT, () => {
44    |   console.log(`Server is running on port ${PORT}.`);
45  | });
```

Fig.5 - Server.js


```

app > config > JS db.config.js > [e] <unknown>
1  module.exports = {
2      HOST: process.env.RDS_HOSTNAME,
3      PORT: process.env.RDS_PORT,
4      DEFAULT_DB: process.env.RDS_DB_NAME,
5      COMMUNITY_DB: process.env.RDS_COMMUNITY_DB,
6      USER: process.env.RDS_USERNAME,
7      PASS: process.env.RDS_PASSWORD,
8      DIALECT: "mysql",
9      pool: {
10         max: 8,
11         min: 0,
12         acquire: 30000,
13         idle: 10000
14     }
15 };
16

```

db.config.js

```

app > controllers > JS appointment.controller.js > ...
1  const { condition } = require("sequelize");
2  const db = require("../models");
3  const Appointment = db.appointment;
4  const Op = db.Sequelize.Op;
5
6  // Create and Save a new Data
7  exports.create = (req, res) => {
8
9      const {
10         appointmentid,
11         paintentid,
12         roomid,
13         departmentid,
14         appointmentdate,
15         appointmenttime,
16         doctorid,
17         status,
18         appoinmenttype,
19         app_reason,
20         ipd,
21         consultancy,
22         payment_mode,
23         payment,
24         reception_time,
25         doctor_time,
26         doctor_outtime,
27         pharma_time
28     } = req.body
29
30     const data1 = {
31         appointmentid,
32         paintentid,
33         roomid,
34         departmentid,
35         appointmentdate,
36         appointmenttime,
37         doctorid,
38         status,
39         appoinmenttype,
40         app_reason,
41         ipd,
42         consultancy,
43         payment_mode,
44         payment

```

app > controllers > JS appointment.controller.js > ...

```
44     payment,
45     reception_time,
46     doctor_time,
47     doctor_outtime,
48     pharma_time
49   };
50
51   // Save Address in the database
52   return Appointment.create(data1)
53     .then(async data => {
54       data = data.get({ plain: true });
55
56       return res.status(201).json({
57         code: 200,
58         message: "Appointment added successfully",
59         data
60       });
61     })
62     .catch(err => {
63       console.error(`Error Creating Appointment :: ${err}`);
64       return res.status(500).json({ message: 'Internal Server Error'
65     });
66   });
67
68
69   // Retrieve all Address from the database.
70   exports.findAll = (req, res) => {
71
72     const limit = Math.min(parseInt(req.query.limit || 10), 10);
73
74     const condition = {}
75
76     if (req.query.department) {
77       condition.department = req.query.department
78     }
79
80     if (req.query.nextToken) {
81       condition.updatedAt = { [Op.lt]: new Date(req.query.nextToken) };
82     }
83
84     return Appointment.findAll({
85       where: condition,
86       order: [['updatedAt', 'desc']],
87       limit: limit,
```

app > controllers > JS appointment.controller.js > ...

```
188     raw: true,
189   })
190   .then(async data => {
191
192     const nextToken = data.length === limit ? data[limit - 1].updateToken : null;
193
194     return res.status(200).json({ data, nextToken });
195   })
196   .catch(err => {
197     console.error(`Error Getting Addresses :: ${err}`);
198     return res.status(500).json({ message: "Internal Server Error" });
199   });
200
201
202
203
204 exports.deleteAll = (req, res) => {
205
206
207   return Appointment.destroy({
208     where: {
209       experience: {
210         [Op.gte]: 20
211       }
212     }
213   })
214   .then(async data => {
215
216     return res.status(200).json({ message: "Appontiment deleted successfully" });
217   })
218   .catch(err => {
219     console.error(`Error deleteing employee :: ${err}`);
220     return res.status(500).json({ message: "Internal Server Error" });
221   });
222
223 };
224
225
226
227
```

appointment.controller.js


```

app > middlewares > JS authMiddleware.js > ...
1  const jwt = require("jsonwebtoken");
2  const asyncHandler = require("express-async-handler");
3  const db = require("../models");
4  const User = db.user;
5
6  const Op = db.Sequelize.Op;
7  const bypass = ['/api/notification/device', '/api/pagewidget'];
8
9
10 const protect = asyncHandler(async (req, res, next) => {
11   if (
12     req.headers.authorization &&
13     req.headers.authorization.startsWith("Bearer")
14   ) {
15     try {
16       const token = req.headers.authorization.split(" ")[1];
17       const decoded = jwt.verify(token, process.env.JWT_SECRET);
18
19       req.user = await User.findOne({
20         where: { id: decoded.id },
21         raw: true,
22       });
23
24     } catch (error) {
25       const tokenExpired = error.name === 'TokenExpiredError';
26       console.error(`Error Authenticating Token :: ${error}`);
27       return res.status(401).json({ message: "Unauthorised", tokenExp
28     }
29   }
30
31   if ((!req.user || !req.user.id) && !bypass.includes(req.originalUrl.spl
32     return res.status(401).json({ message: "Unauthorised" });
33   }
34
35   next();
36 });
37
38
39
40 const admin = (req, res, next) => {
41   if (req.user && req.user.role === "ADMIN") {
42     next();
43   } else {
44     return res.status(401).json({ message: "Unauthorised" });

```

```

45   }
46 };
47
48
49 module.exports = { protect, admin };
50

```

authMiddleware.js

app > models > JS appointment.model.js > <unknown>

```
1 module.exports = (sequelize, Sequelize) => {
2   ⚡ const appointment = sequelize.define("appointment", {
3     appointmentid: {
4       type: Sequelize.INTEGER.UNSIGNED,
5       autoIncrement: true,
6       primaryKey: true
7     },
8     patientid: {
9       type: Sequelize.INTEGER.UNSIGNED,
10      allowNull: false
11    },
12    roomid: {
13      type: Sequelize.INTEGER.UNSIGNED,
14      allowNull: false
15    },
16    departmentid: {
17      type: Sequelize.INTEGER.UNSIGNED,
18      allowNull: false
19    },
20    appointmentdate: {
21      type: Sequelize.DATE,
22      allowNull: false
23    },
24    appointmenttime: {
25      type: Sequelize.TIME,
26      allowNull: false
27    },
28    doctorid: {
29      type: Sequelize.INTEGER.UNSIGNED,
30      allowNull: false
31    },
32    status: {
33      type: Sequelize.STRING(10),
34      allowNull: false
35    },
36    appointmenttype: {
37      type: Sequelize.STRING(25),
38      allowNull: false
39    },
40    app_reason: {
41      type: Sequelize.TEXT,
42      allowNull: false
43    },
44    ipd: {
```

```

app > models > JS appointment.model.js > <unknown>
44     ipd: {
45         type: Sequelize.INTEGER.UNSIGNED,
46         allowNull: false
47     },
48     consultancy: {
49         type: Sequelize.INTEGER.UNSIGNED,
50         allowNull: false
51     },
52     payment_mode: {
53         type: Sequelize.STRING(50),
54         allowNull: false
55     },
56     payment: {
57         type: Sequelize.INTEGER.UNSIGNED,
58         allowNull: false
59     },
60     reception_time: {
61         type: 'TIMESTAMP',
62         defaultValue: Sequelize.literal('CURRENT_TIMESTAMP'),
63         allowNull: false
64     },
65     doctor_time: {
66         type: 'TIMESTAMP',
67         defaultValue: Sequelize.literal('CURRENT_TIMESTAMP'),
68         allowNull: false
69     },
70     doctor_outtime: {
71         type: 'TIMESTAMP',
72         defaultValue: Sequelize.literal('CURRENT_TIMESTAMP'),
73         allowNull: false
74     },
75     pharma_time: {
76         type: 'TIMESTAMP',
77         defaultValue: Sequelize.literal('CURRENT_TIMESTAMP'),
78         allowNull: false
79     }
80 }, {
81     freezeTableName: true
82 });
83
84 return appointment;
85 };
86

```

appointment.model.js

```

app > routes > JS appointment.routes.js > ...
1  const appointment = require("../controllers/appoinment.controller");
2  const { appointment } = require("../models");
3
4  module.exports = app => {
5      const router = require("express").Router();
6      router.post("/", appointment.create);
7      app.use("/api/appoinment", router);
8  };

```

appointment.routes.js