**Module 72-Final Project Part-2 (Appoinment)**

72-1 Module Introduction, Recap and Appointment Page

#### 72-2 Create Appointment Banner using React Day Picker

**Day Picker :**

<https://react-day-picker.js.org/start>

import React, { useState } from 'react';

import bgChair from "../../assets/images/bg.png";

import chair from "../../assets/images/chair.png";

import { DayPicker } from "react-day-picker";

import "react-day-picker/dist/style.css";

const AppointmentBanner = () => {

    const [date,setDate]=useState(new Date());

    return (

      <div className="hero min-h-screen">

        <img src={bgChair} className="mt-[-100px]" alt="" />

        <div className="hero-content flex-col lg:flex-row-reverse ">

          <img

            src={chair}

            className="max-w-sm rounded-lg shadow-2xl w-50 ml-10"

            alt="Dentice Chair"

          />

          <div className="">

            <DayPicker mode="single" selected={date} onSelect={setDate} />

          </div>

        </div>

      </div>

    );

};

export default AppointmentBanner;

#### 72-3 Lift up state to share data and Available Appointment Component

#### 72-4 Booking Options and available slots with conditional rendering

#### 72-5 Display a modal from on booking click with dynamic data

#### 72-6 Create Booking form inside Booking modal

#### 72-7 Simple Node Mongo Server for doctors portal

const express = require('express');

const app = express();

const cors = require("cors");

require("dotenv").config();

const { MongoClient, ServerApiVersion } = require("mongodb");

const port = process.env.PORT || 5000;

// midleware

app.use(cors());

app.use(express.json());

const uri = `mongodb+srv://${process.env.DB\_USER}:${process.env.DB\_PASS}@cluster0.dilsw.mongodb.net/myFirstDatabase?retryWrites=true&w=majority`;

const client = new MongoClient(uri, {

  useNewUrlParser: true,

  useUnifiedTopology: true,

  serverApi: ServerApiVersion.v1,

});

client.connect((err) => {

  const collection = client.db("test").collection("devices");

  // perform actions on the collection object

  console.log("connect");

  client.close();

});

app.get("/", (req, res) => {

  res.send("Hello From Doctor Uncle!");

});

app.listen(port, () => {

  console.log(`Doctors App listening on port ${port}`);

});

#### 72-8 Connect Mongodb and load services from database

#### 72-9 Module summary