Codes

**Header.ejs**

<!doctype html>

<html lang="en">

<head>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"

        integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

    <link rel="stylesheet" href="../css/style.css">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css"

        integrity="sha512-KfkfwYDsLkIlwQp6LFnl8zNdLGxu9YAA1QvwINks4PhcElQSvqcyVLLD9aMhXd13uQjoXtEKNosOWaZqXgel0g=="

        crossorigin="anonymous" referrerpolicy="no-referrer" />

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

    <title>Course Website</title>

</head>

<body>

    <header>

        <h1><a href="../"> Computer Science Learning Resources</a></h1>

        <a class="drive-button"

            href="https://drive.google.com/drive/folders/1qCeBUHSXqH5xWX0kh2UbXI1ymlj6Qut\_?usp=sharing"

            target="\_blank"><span> Get G-Drive Access</span></a>

        <a class="drive-button" href="/add"><span>Add New Book</span></a>

    </header>

**Footer.ejs**

<div class="footer-dark">

    <footer>

        <div class="container">

            <div class="row">

                <div class="col item social">

                    <a href="https://www.facebook.com/profile.php?id=100008773902927" target="\_blank"><i class="fab fa-facebook-f"></i></a>

                    <a href="https://instagram.com/lovely\_sharma0812" target="\_blank"><i class="fab fa-instagram"></i></a>

                    <a href="https://github.com/CoderLovely08" target="\_blank"><i class="fab fa-github"></i></a>

                    <a href="https://linkedin.com/in/lovely-sharma-a54022214" target="\_blank"><i class="fab fa-linkedin"></i></a>

                    <a href="https://t.me/CoderLovely08" target="\_blank"><i class="fab fa-telegram"></i></a>

                    <br>

                    <!-- <br> -->

                        <h4>Connect with me on Social Media</h4>

                </div>

                <div class="col-md-6 item text">

                    <h5>Click <a href="https://github.com/CoderLovely08/study-resource-website">here </a>to make a contribution to this project and imporve the existing UI.</h5>

                </div>

            </div>

            <p class="copyright">Creator Lovely Sharma © 2022</p>

        </div>

    </footer>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.js"></script>

<script src="script/script.js"></script>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"

    integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"

    crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"

    integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"

    crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"

    integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"

    crossorigin="anonymous"></script>

</body>

</html>

**Home.ejs**

<%- include("partials/header") -%>

    <section>

        <div class="resourceContainer">

            <% for(let i=0; i<books.length;i++){ %>

                <div class="resource-item">

                    <a href=<%="/posts/" + books[i].book\_route %> target="\_blank">

                        <img src=<%=books[i].book\_image\_src %> alt=""><br>

                        <%= books[i].book\_title %>

                    </a>

                </div>

            <% } %>

        </div>

    </section>

    <%- include("partials/footer") -%>

**App.js**

//jshint esversion:6

const express = require("express");

const bodyParser = require("body-parser");

const ejs = require("ejs");

const imgur = require('imgur-uploader');

const fs = require("fs")

const fileupload = require("express-fileupload");

const loadsh = require("lodash")

const session = require('express-session');

require("dotenv").config();

var async = require('async');

// For MySQL database connection

var mysql = require('mysql');

var client = mysql.createConnection({

    host: "sql12.freemysqlhosting.net",

    user: "sql12561563",

    password: "MAkRFeRh3z",

    database: "sql12561563"

});

client.connect(function (err) {

    if (err) throw err;

    console.log("Connected!");

});

// const { Client } = require("pg");

const oneDay = 1000 \* 60 \* 60 \* 24;

const app = express();

app.set('view engine', 'ejs');

app.use(bodyParser.urlencoded({ extended: true }));

app.use(express.static("public"));

app.use(fileupload());

app.use(session({

    secret: "thisismysecrctekeyfhrgfgrfrty84fwir767",

    saveUninitialized: true,

    cookie: { maxAge: oneDay },

    resave: true,

    withCredential : true

}));

/\*-------------------------------------------

                Global Variables

  -------------------------------------------\*/

// const client = new Client({

//     host: process.env.ADMIN\_HOST,

//     user: process.env.ADMIN\_USER,

//     port: process.env.ADMIN\_PORT,

//     password: process.env.ADMIN\_PASSWORD,

//     database: process.env.ADMIN\_DATABASE,

//     idleTimeoutMillis: 0,

//     connectionTimeoutMillis: 0

// });

// client.connect();

/\*-------------------------------------------

                Home route

  -------------------------------------------\*/

app.get("/", function (req, res) {

    // Querying all records to display all books

    client.query("Select \* from BookData order by book\_title", (err, results) => {

        console.log(results);

        res.render("home", { books: results });

    });

});

/\*-------------------------------------------

                Add Book Route

  -------------------------------------------\*/

//   This route renders the admin panel for adding new books

app.get("/add", function (req, res) {

    sess = req.session;

    if (sess.isLoggedin) {

        res.render("add");

    } else {

        res.render('admin');

    }

});

app.post("/admin", function (req, res) {

    sess = req.session;

    let username = req.body.username;

    let adminpass = req.body.adminPassword;

    let query = `Select \* from AdminInfo where username = '${username}' and login\_password= '${adminpass}'`;

    client.query(query, (err, results) => {

        if (results.length == 1) {

            req.session.isLoggedin = results[0].username;

            req.session.adminId = results[0].admin\_id;

            res.render("add");

        } else {

            res.render('admin');

        }

    });

});

/\*-------------------------------------------

                Upload Route

  -------------------------------------------\*/

app.post("/upload", function (req, res) {

    // check if files are not empty

    if (!req.files) {

        return res.status(400).send("No files Found!");

    }

    // if file exists store it in myfile variable

    let myfile = req.files.thumbnail;

    let uploadPath = \_\_dirname + "/uploads/" + myfile.name;

    // move the file to uploads folder for temp storage

    myfile.mv(uploadPath, function (err) {

        // if (err) console.log("Error!");

        // upload the moved file to imgur and recieve a callback

        imgur(fs.readFileSync(uploadPath)).then(data => {

            // set the attributes based on the response we get

            let bookTitle = req.body.bookTitle;

            let bookRoute = req.body.bookRoute.trim();

            let imageSrc = data.link;  // only this is important for us

            let folderUrl = req.body.bookLink;

            let references = req.body.bookReferences;

            // Check route if exists

            async.series([

                function (callback) {

                    let checkquery = `Select \* from BookData where book\_route = '${bookRoute}'`;

                    client.query(checkquery, (err, results) => {

                        if (results.length != 0) {

                            callback(true);

                        }

                        else callback();

                    });

                },

                function (callback) {

                    // inserting the data into our book database

                    let query = `Insert into BookData(book\_title, book\_route, book\_image\_src, book\_folderlink) values('${bookTitle}','${bookRoute}','${imageSrc}','${folderUrl}')`;

                    client.query(query, (err, result) => {

                        callback();

                    });

                },

                function (callback) {

                    // Inserting references if any by selecting the latest id

                    if (references.trim().length != 0) {

                        console.log("inside references");

                        query = "Select \* from BookData order by book\_id desc Limit 1";

                        let latestBookId;

                        client.query(query, (err, result) => {

                            latestBookId = result[0].book\_id;

                            // If admin has provided any reference then insert them all

                            if (references.length != 0 && latestBookId != -1) {

                                let insertQuery = `insert into BookReferences(book\_id,book\_reference\_name,book\_reference\_link) values`

                                let newItem = references.split(",");

                                for (let i = 0; i < newItem.length; i++) {

                                    let eachItem = newItem[i].split("http");

                                    eachItem[1] = "http" + eachItem[1];

                                    let refName = eachItem[0].trim();

                                    let refLink = eachItem[1].trim();

                                    insertQuery = insertQuery + `(${latestBookId},'${refName}','${refLink}')`

                                    if (i != newItem.length - 1) insertQuery += ","

                                }

                                client.query(insertQuery, (err, result) => {

                                    if (err) console.log(err);

                                });

                            }

                            callback();

                        });

                    }

                }

            ],

                function (err, result) {

                    if (err) { res.render("error"); }

                    // redirect admin to the home after inserting and display new updated book

                    if (result) res.redirect("/");

                });

            // removing the file from our temporary uploads folder

            fs.unlinkSync(uploadPath);

        });

    });

});

/\*-------------------------------------------

                Display Book Route

  -------------------------------------------\*/

app.get("/posts/:postId", function (req, res) {

    // perform linear search for the route

    // if found just render that page only

    let sqlQuery = "Select BookData.book\_id,BookData.book\_title,BookData.book\_route,BookData.book\_image\_src,BookData.book\_folderlink,AdminInfo.admin\_name from BookData,AdminInfo where BookData.admin\_id=AdminInfo.admin\_id";

    client.query(sqlQuery, (err, results) => {

        let booksData = results;

        let bookTitle = loadsh.lowerCase(req.params.postId);

        for (let i = 0; i < booksData.length; i++) {

            if (bookTitle === booksData[i].book\_route) {

                client.query(`Select \* from BookReferences where book\_id = ${booksData[i].book\_id}`, (err, refResult) => {

                    // console.log(refResult);

                    res.render("posts", { bookData: booksData[i], bookReferences: refResult });

                });

                break;

            }

        }

    });

});

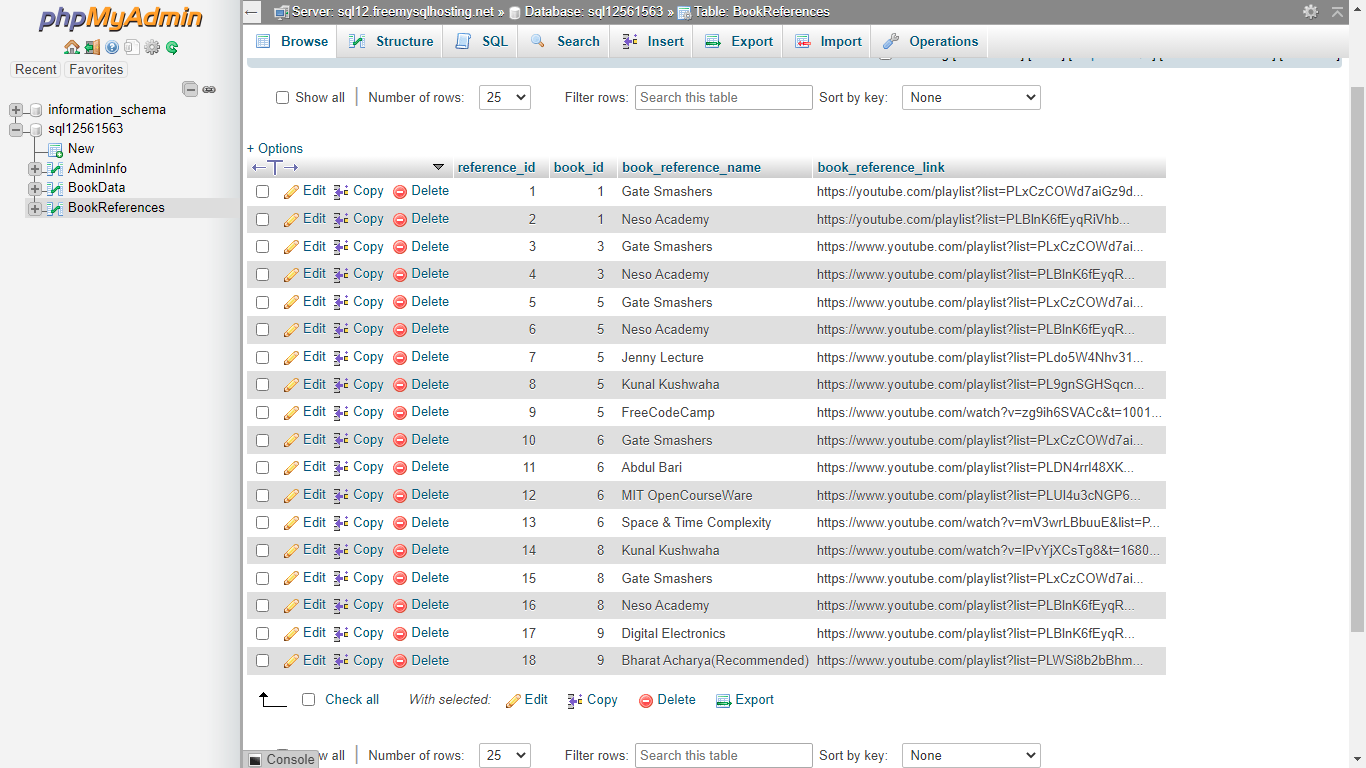
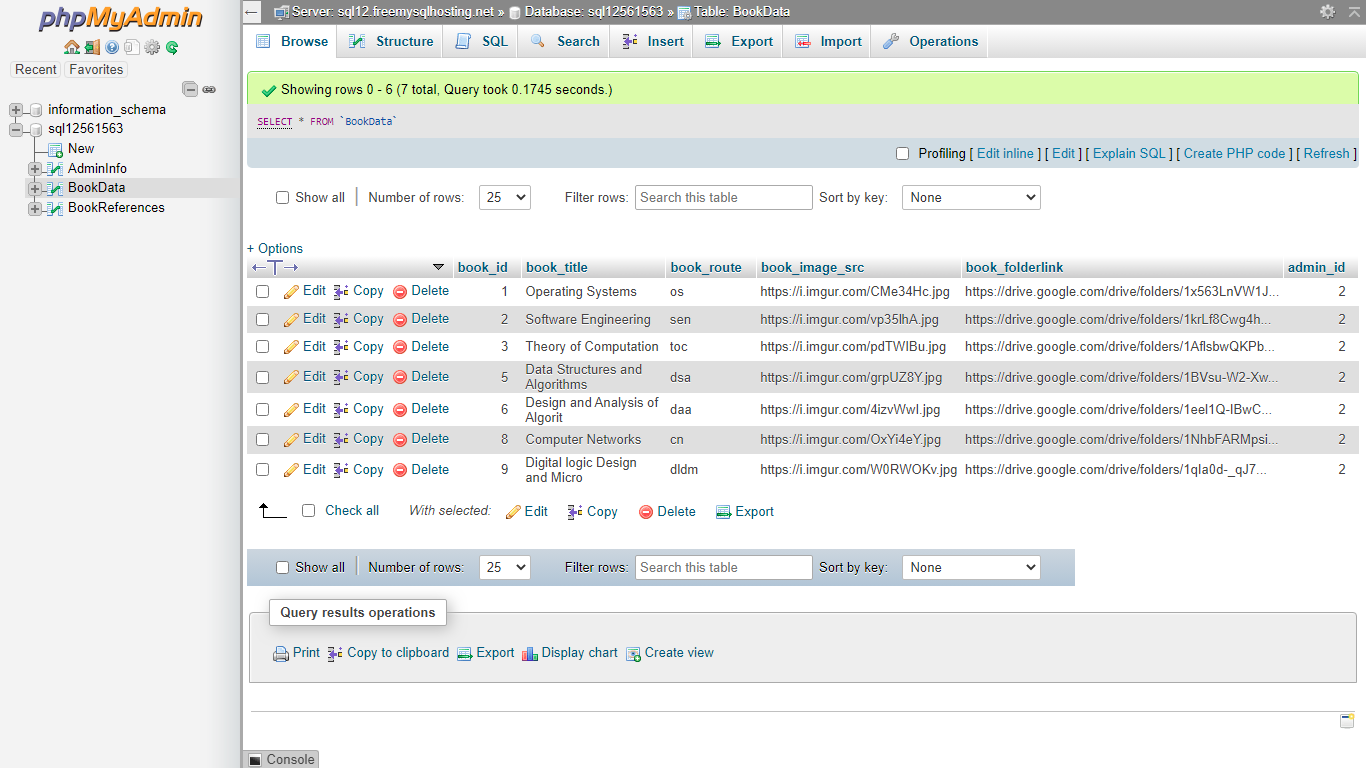
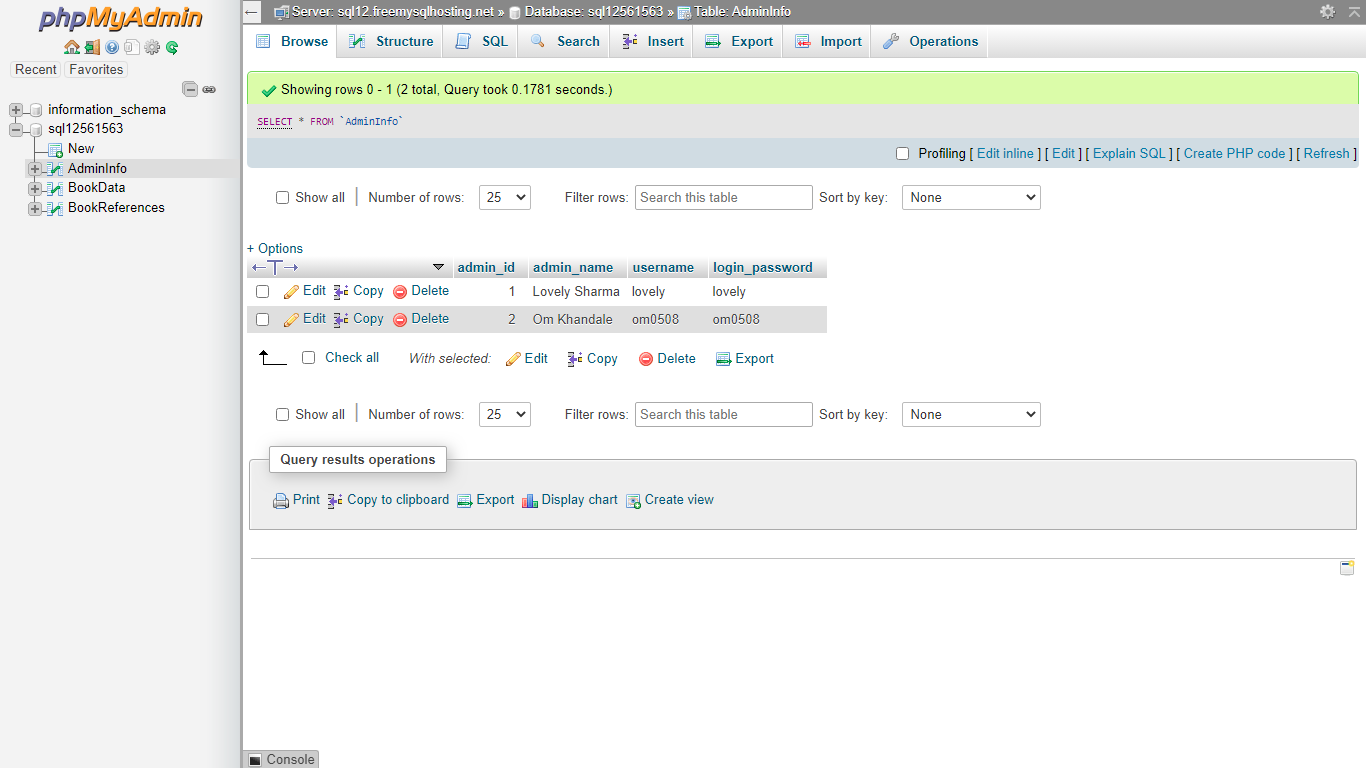
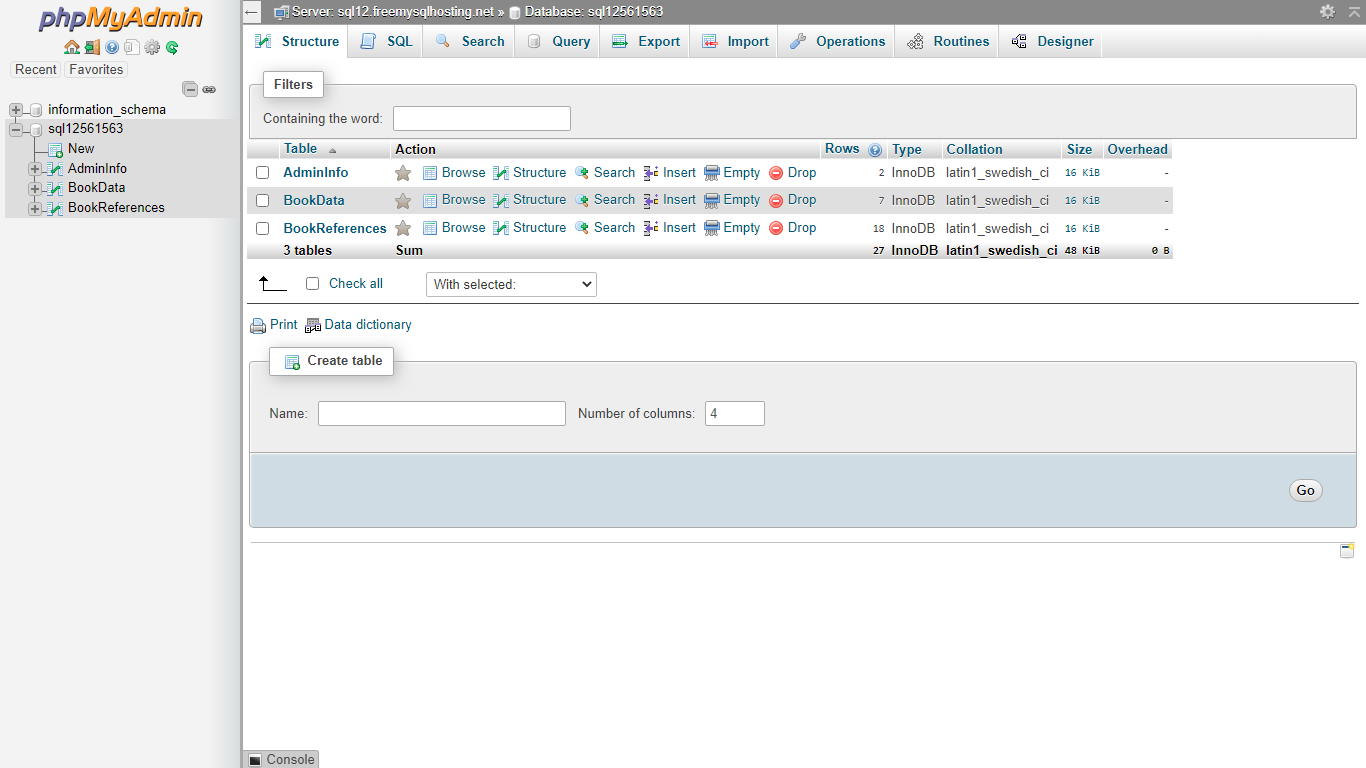
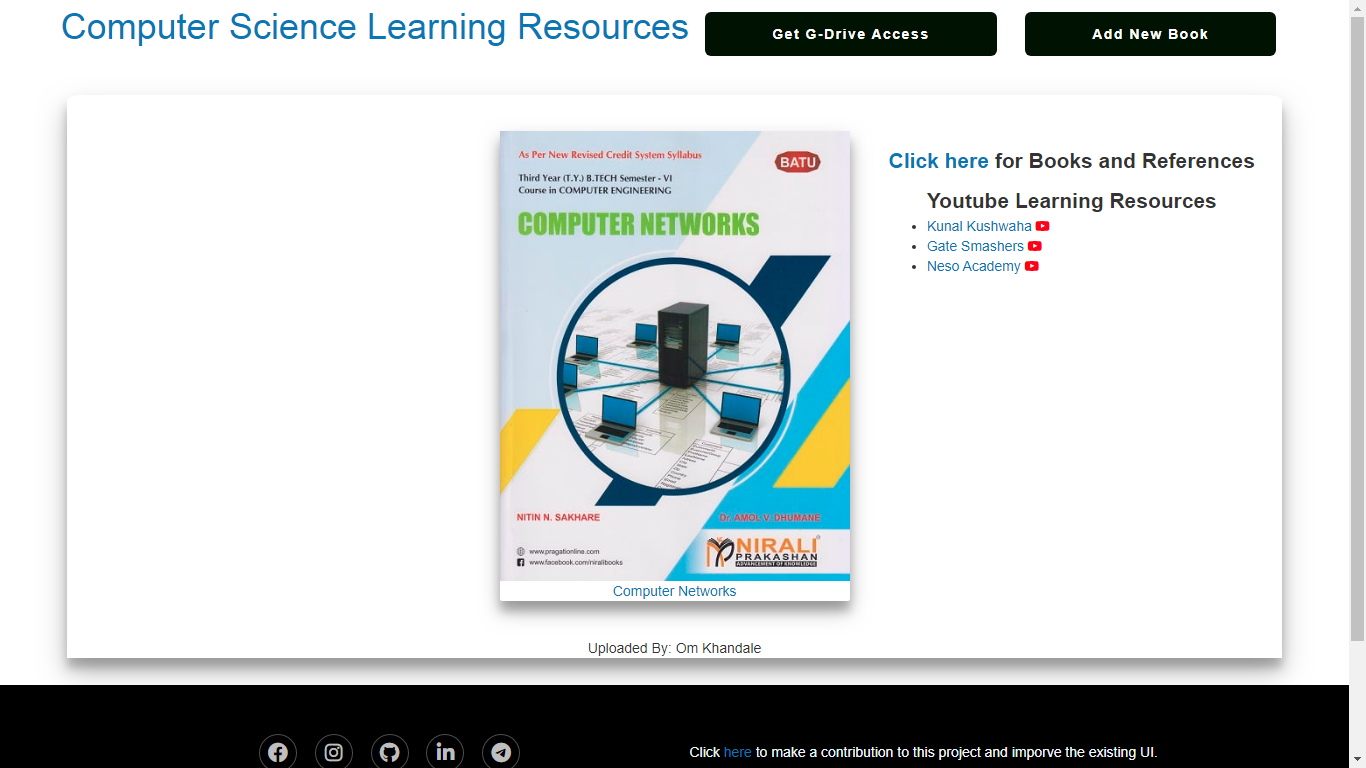
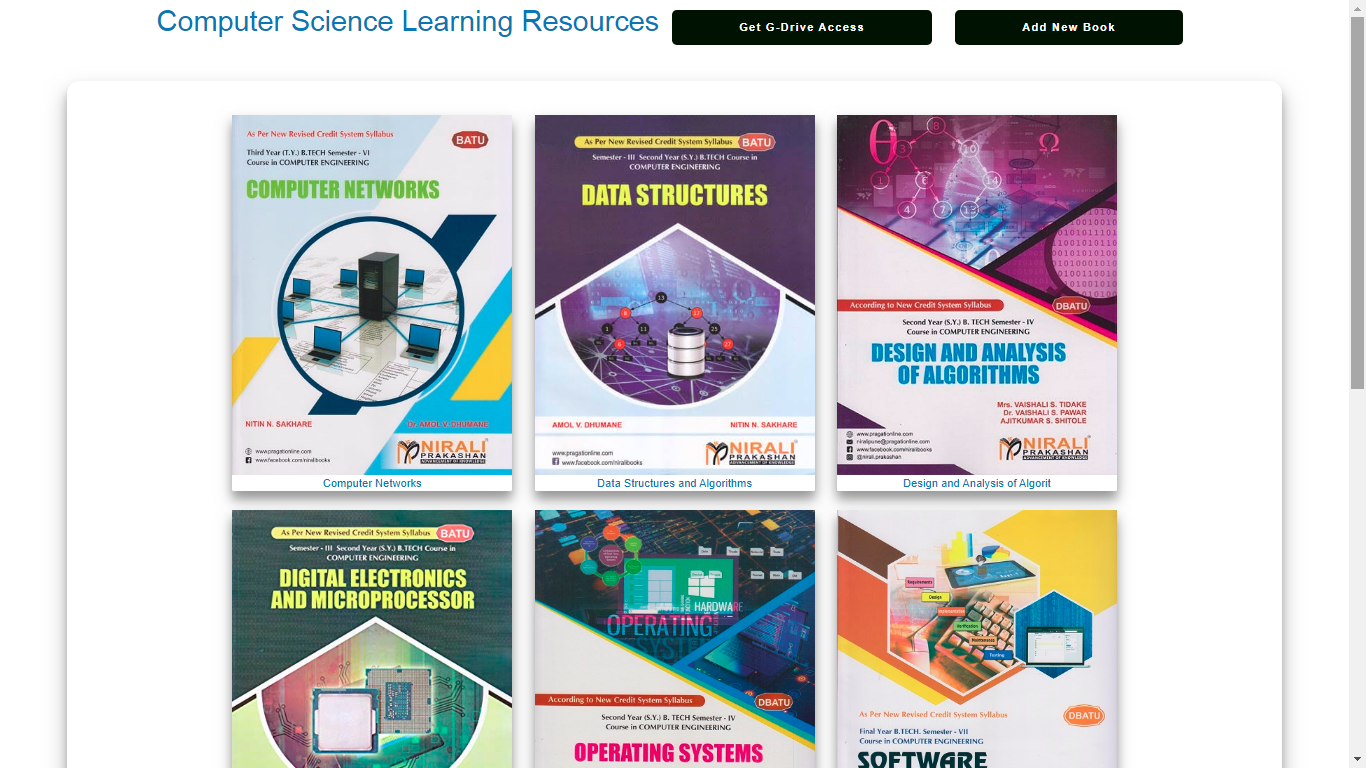
// Starting the app

app.listen(process.env.PORT || 3000, function () {

    console.log("Server is running on port 3000!");

});

Snapshots



SQL Files

-- --------------------------------------------------------

--

-- Table structure for table `AdminInfo`

--

DROP TABLE IF EXISTS `AdminInfo`;

CREATE TABLE `AdminInfo` (

  `admin\_id` int(11) NOT NULL,

  `admin\_name` varchar(30) DEFAULT NULL,

  `username` varchar(20) DEFAULT NULL,

  `login\_password` varchar(30) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `AdminInfo`

--

INSERT INTO `AdminInfo` (`admin\_id`, `admin\_name`, `username`, `login\_password`) VALUES

(1, 'Lovely Sharma', 'lovely', 'lovely'),

(2, 'Om Khandale', 'om0508', 'om0508');

-- --------------------------------------------------------

--

-- Table structure for table `BookData`

--

DROP TABLE IF EXISTS `BookData`;

CREATE TABLE `BookData` (

  `book\_id` int(3) NOT NULL,

  `book\_title` varchar(30) DEFAULT NULL,

  `book\_route` varchar(10) DEFAULT NULL,

  `book\_image\_src` varchar(50) DEFAULT NULL,

  `book\_folderlink` varchar(255) DEFAULT NULL,

  `admin\_id` int(11) DEFAULT '1'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `BookData`

--

INSERT INTO `BookData` (`book\_id`, `book\_title`, `book\_route`, `book\_image\_src`, `book\_folderlink`, `admin\_id`) VALUES

(1, 'Operating Systems', 'os', 'https://i.imgur.com/CMe34Hc.jpg', 'https://drive.google.com/drive/folders/1x563LnVW1JoTaMT3zv9t5t2d8ktgnjM4', 2),

(2, 'Software Engineering', 'sen', 'https://i.imgur.com/vp35lhA.jpg', 'https://drive.google.com/drive/folders/1krLf8Cwg4hQzpuV8fu-KrCjR\_nFgnZPZ', 2),

(3, 'Theory of Computation', 'toc', 'https://i.imgur.com/pdTWIBu.jpg', 'https://drive.google.com/drive/folders/1AflsbwQKPbpNHp1FP\_9TdC3SINcjM3Ta', 2),

(5, 'Data Structures and Algorithms', 'dsa', 'https://i.imgur.com/grpUZ8Y.jpg', 'https://drive.google.com/drive/folders/1BVsu-W2-XwWTxQQNEJU94mcmoQyg94x4', 2),

(6, 'Design and Analysis of Algorit', 'daa', 'https://i.imgur.com/4izvWwI.jpg', 'https://drive.google.com/drive/folders/1eel1Q-IBwCgEnz6zb5XYiztZT5SBY2Q7', 2),

(8, 'Computer Networks', 'cn', 'https://i.imgur.com/OxYi4eY.jpg', 'https://drive.google.com/drive/folders/1NhbFARMpsiip5RYdIyRUNxnB0s-JPKPF?usp=sharing', 2),

(9, 'Digital logic Design and Micro', 'dldm', 'https://i.imgur.com/W0RWOKv.jpg', 'https://drive.google.com/drive/folders/1qIa0d-\_qJ7jyWJZSH045RunCSvAOBYWe?usp=sharing', 2);

-- --------------------------------------------------------

--

-- Table structure for table `BookReferences`

--

DROP TABLE IF EXISTS `BookReferences`;

CREATE TABLE `BookReferences` (

  `reference\_id` int(11) NOT NULL,

  `book\_id` int(11) NOT NULL,

  `book\_reference\_name` varchar(255) NOT NULL,

  `book\_reference\_link` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `BookReferences`

--

INSERT INTO `BookReferences` (`reference\_id`, `book\_id`, `book\_reference\_name`, `book\_reference\_link`) VALUES

(1, 1, 'Gate Smashers', 'https://youtube.com/playlist?list=PLxCzCOWd7aiGz9donHRrE9I3Mwn6XdP8p'),

(2, 1, 'Neso Academy', 'https://youtube.com/playlist?list=PLBlnK6fEyqRiVhbXDGLXDk\_OQAeuVcp2O'),

(3, 3, 'Gate Smashers', 'https://www.youtube.com/playlist?list=PLxCzCOWd7aiFM9Lj5G9G\_76adtyb4ef7i'),

(4, 3, 'Neso Academy', 'https://www.youtube.com/playlist?list=PLBlnK6fEyqRgp46KUv4ZY69yXmpwKOIev'),

(5, 5, 'Gate Smashers', 'https://www.youtube.com/playlist?list=PLxCzCOWd7aiEwaANNt3OqJPVIxwp2ebiT'),

(6, 5, 'Neso Academy', 'https://www.youtube.com/playlist?list=PLBlnK6fEyqRj9lld8sWIUNwlKfdUoPd1Y'),

(7, 5, 'Jenny Lecture', 'https://www.youtube.com/playlist?list=PLdo5W4Nhv31bbKJzrsKfMpo\_grxuLl8LU'),

(8, 5, 'Kunal Kushwaha', 'https://www.youtube.com/playlist?list=PL9gnSGHSqcnr\_DxHsP7AW9ftq0AtAyYqJ'),

(9, 5, 'FreeCodeCamp', 'https://www.youtube.com/watch?v=zg9ih6SVACc&t=10015s'),

(10, 6, 'Gate Smashers', 'https://www.youtube.com/playlist?list=PLxCzCOWd7aiHcmS4i14bI0VrMbZTUvlTa'),

(11, 6, 'Abdul Bari', 'https://www.youtube.com/playlist?list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs\_O'),

(12, 6, 'MIT OpenCourseWare', 'https://www.youtube.com/playlist?list=PLUl4u3cNGP6317WaSNfmCvGym2ucw3oGp'),

(13, 6, 'Space & Time Complexity', 'https://www.youtube.com/watch?v=mV3wrLBbuuE&list=PL9gnSGHSqcnr\_DxHsP7AW9ftq0AtAyYqJ&index=24&t=6960s'),

(14, 8, 'Kunal Kushwaha', 'https://www.youtube.com/watch?v=IPvYjXCsTg8&t=1680s'),

(15, 8, 'Gate Smashers', 'https://www.youtube.com/playlist?list=PLxCzCOWd7aiGFBD2-2joCpWOLUrDLvVV\_'),

(16, 8, 'Neso Academy', 'https://www.youtube.com/playlist?list=PLBlnK6fEyqRgMCUAG0XRw78UA8qnv6jEx'),

(17, 9, 'Digital Electronics', 'https://www.youtube.com/playlist?list=PLBlnK6fEyqRjMH3mWf6kwqiTbT798eAOm'),

(18, 9, 'Bharat Acharya(Recommended)', 'https://www.youtube.com/playlist?list=PLWSi8b2bBhm7bKvNPOln-sghqiHUcnH7V');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `AdminInfo`

--

ALTER TABLE `AdminInfo`

  ADD PRIMARY KEY (`admin\_id`);

--

-- Indexes for table `BookData`

--

ALTER TABLE `BookData`

  ADD PRIMARY KEY (`book\_id`);

--

-- Indexes for table `BookReferences`

--

ALTER TABLE `BookReferences`

  ADD PRIMARY KEY (`reference\_id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `AdminInfo`

--

ALTER TABLE `AdminInfo`

  MODIFY `admin\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT for table `BookData`

--

ALTER TABLE `BookData`

  MODIFY `book\_id` int(3) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=10;

--

-- AUTO\_INCREMENT for table `BookReferences`

--

ALTER TABLE `BookReferences`

  MODIFY `reference\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=31;