



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

The BookStore: Access To College Material During Pandemic

¹Varun Shinde, ²Sruthisri Venkateswaran, ³Prapthi Shetty, ⁴Pranita Mahajan

¹Student, ²Student, ³Student, ⁴Assistant Professor

¹Computer Engineering,

¹SIES Graduate School of Technology, Navi Mumbai, India

Abstract: This web application is based on MVC Architecture. The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle distinct development aspects of an application. This web application is built using MERN stack, and it helps to generate e-books using notes provided in images format or any kind of google documents (using features of Google Drive for required documents). The e-books created are then accessible by every student of the college and helps them bridge the gap between classroom and distance teaching. The notes in image format are stored in S3 cloud services in a compressed manner to avoid usage of server space. The Google SERP API used helps to generate automated recommendations for related subjects to improve the learning experience.

Index Terms -E Books, MVC architecture, S3, MERN stack, Google SERP

I. INTRODUCTION

The COVID-19 pandemic put an unforeseen stop to academic matters and also on classroom studies and libraries. Within days, shops were closed for straight months and without enough faculty members to whom students could approach to solve their problems, students realized they were lacking the required resources. The only resources they had were the notes they had taken down during the lecture hours and also some books if they had them at home.

So, keeping this challenge in mind, the members of this project got to work on it. With this project, the members wanted every student to get equal opportunity regardless of where they were and whether they had resources or not. This project ensured that no one was left behind.

The main challenge in front of the developers was to organize the notes given by their teacher in a better way because they could get lost in the endless clutter of the PDFs on their phones and laptops. Also, students were sending these handwritten notes written by their teachers prepared for their college students to other colleges, which can be viewed as piracy. Hence, to solve these, the developers came up with this website i.e., The BookStore

II. LITERATURE SURVEY

Although there are many services available that provide some sort of scanned reference books, none of them could cater to the custom demands of the college environment and also the cost factor. This gave us ample space to work on our project. For better management, we have used Amazon Web Services for the storage of files and the application is hosted on Heroku. The website is for all four years of computer engineering.

Google Search API has also been added to help students with their studies and get all topics in one single place. The User Interface of the site is very user-friendly and attractive to the eyes, leading to better visuals. The teachers have been given many more options on the page exclusive to them along with an admin panel and the ability to edit books that they upload, giving them full control over the books that they will upload. Sites like GeeksForGeeks[7], Javatpoint[8], Tutorialspoint[9], are amongst the many websites that provide students with subject knowledge of their fields. Their contents can be accessed freely by any student, irrespective of the university or college they belong to.

The limitation of this system is that the members not only wanted to upload PDF of our teachers' notes but also of reputed publications online because students in this type of pandemic cannot go to libraries, also buying books in this pandemic was tough, and it requires a lot of money, this was the factor that made us only focus on the localized version i.e. getting teacher's notes and not anything else. Being a college focused application, it had to be very easy to

use because otherwise many students would refrain from using it, also it should encourage students to refer to the teacher's notes rather than referring to other online available material which was not up to the standard. It was important to help the teachers to get the best of the system and to connect maximum with the students. If required the project can be expanded to much more boundaries, but this can also lead to much more complex system.

III. RESEARCH MODEL

The website is written in HTML, CSS, and JavaScript. This project has used various libraries such as ExpressJS, Multer, PassportJS and also, and it has used the Figma Design Tool for designing the user interface. The processing system of the project is very simple. It has a node server that controls every request that is sent in the project. The database used is Mongo DB which provides a database for notes uploaded and other details, authentication of emails ID is carried out so that only students and faculty can log in to the website. It is implemented using a Regex code as well as using Google and Microsoft authentication. The file data of the website is stored in the AWS cloud. Depending on the mail ID, the user is classified into 3 roles namely, the faculty, the students, and the admins. The website would be viewed differently for all three roles of users.

Initially, a survey was taken from the entire college community on the quality of online education. Based on this report, it was concluded that learning resources were scarce among the students studying from remote locations. This made learning location dependent. To bridge this gap between online learning and students, we decided to develop an e-learning platform that would help students to learn from any part of the world despite their location. The main objective of this project is to provide quality educational resources through a student-friendly platform and help them excel in their respective fields.

The architecture used in this project is MVC ARCHITECTURE.

Model View Controller (MVC) as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts –

- **Model** – The lowest level of the pattern is responsible for maintaining data.
- **View** – This is responsible for displaying all or a portion of the data to the user.
- **Controller** – Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports the separation of concerns. Here the Controller receives all requests for the application and then works with the model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.

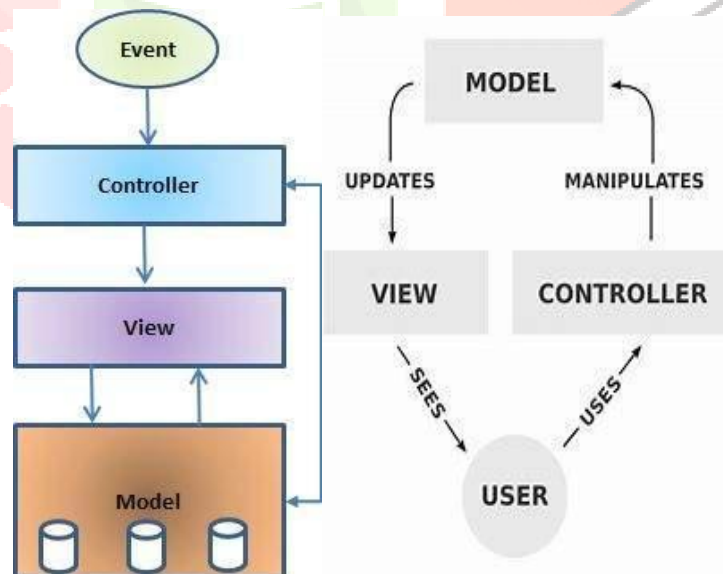


Figure 1

The Model

The model is responsible for managing the data of the application. It responds to the request from the view and responds to the controller's instructions to update itself.

The View

It means the presentation of data in a particular format, triggered by a controller's decision to present the data. They are script-based templating systems like EJS, JSP, ASP, PHP and are very easy to integrate with AJAX technology.

The Controller

The controller is responsible for responding to the user input and performing interactions on the data model objects. The controller receives the input; it validates the input and then performs the business operation that modifies the state of the data model. **ExpressJS** is an MVC-based framework.

This project's processing system is relatively simple. It features a node server that manages all requests sent via the project. It features a database called mongo DB, which Pro demonstrates can be used to store photos and other information. To authenticate emails, this project used Regex code, as well as Google and Microsoft authentication, so that only students and teachers could log in to the website. As the members used so many packages, it has the NPM package manager. Also, the major library that is used is the expressJS library, to keep the project working together GitHub approved an important part because as this time was work from home, that was the only way of sharing code and maintaining a version control effectively.

The hardware and software used in this project are listed below

- HTML
- NodeJS
- MongoDB
- DESIGN SOFTWARE: FIGMA

ARCHITECTURE

How to add books:

Step 1: The teacher has to log in using email ID provided from the college

Step 2: Click on add book on the navigation bar, which will redirect the teacher to the add book page

Step 3: They will be asked to enter the details of the book like the book title, year, subject name, book description which will be followed by uploading the book as either an image or they can provide a PDF link.

Step 4a: If images are uploaded, the images will be stored in the database in the order they were selected while uploading.

Step 4b: If the PDF link is uploaded, it will be saved as it is in the database.

Step 5: Using the image or PDF uploaded by faculties, the e-book will be generated in the database and then will be displayed on the website for everyone to avail in the form of a carousel.

IV.RESULTS AND ANALYSIS

This section explains the functionalities of developed websites with images of respective web pages.

1.HOMEPAGE

The homepage is the first page that is visible to the user as soon as (s)he visits the website. This navbar navigates users to the webpage consisting of all the books by clicking on "Books", "Signup" allows users to signup using either their college Microsoft account or their college Google Mail account. If the user is a faculty, (s)he will be able to add a book by clicking on "Add Book" or edit a book using the "Edit Book" option. The admin gets an extra option that connects him to the dashboard page of the site. The Homepage also gives details about the developers of this project, and the "contact us" section provides ways to contact the developers of the site.

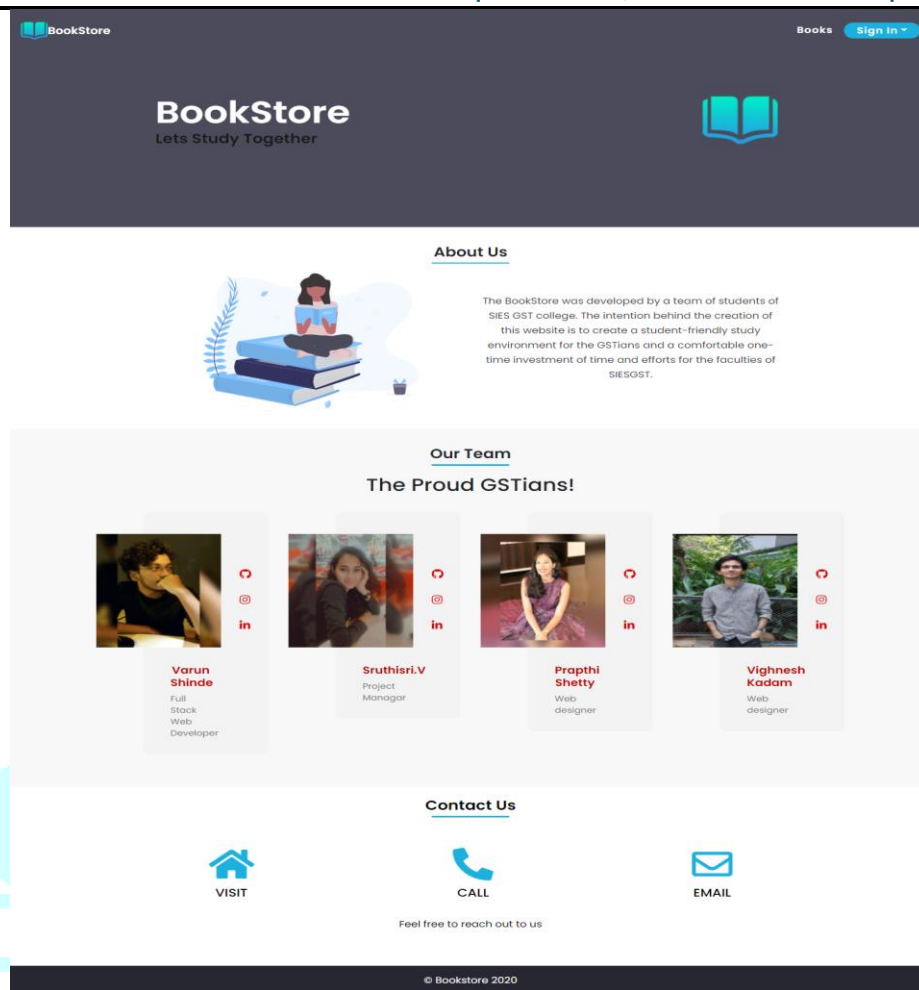


Figure 2

2.ADD BOOKS

Only the admin and the faculty members have access to use the add book section. In this section, they can add books in the format given below. They can upload notes by using Google Drive link as well as they can add notes by directly uploading images of notes.

Figure 3

3.EDIT BOOKS

Suppose a faculty wants to edit an already uploaded book or image, they can do it through this page. Admin or the faculty may be redirected to this page from the Books page.



Edit Book

Book Title:

Select Year:

Select Subject:

Book Description

☒ Use PDF Link
PDF Link

Figure 4



4. BOOKS

Here, the user can search the subject which he wants to study and all the available instances of that subject will be seen in the results. The user can also select the hamburger menu button on the left where he can select if he wants results from a specific year as in the first year, second, or third year of engineering.

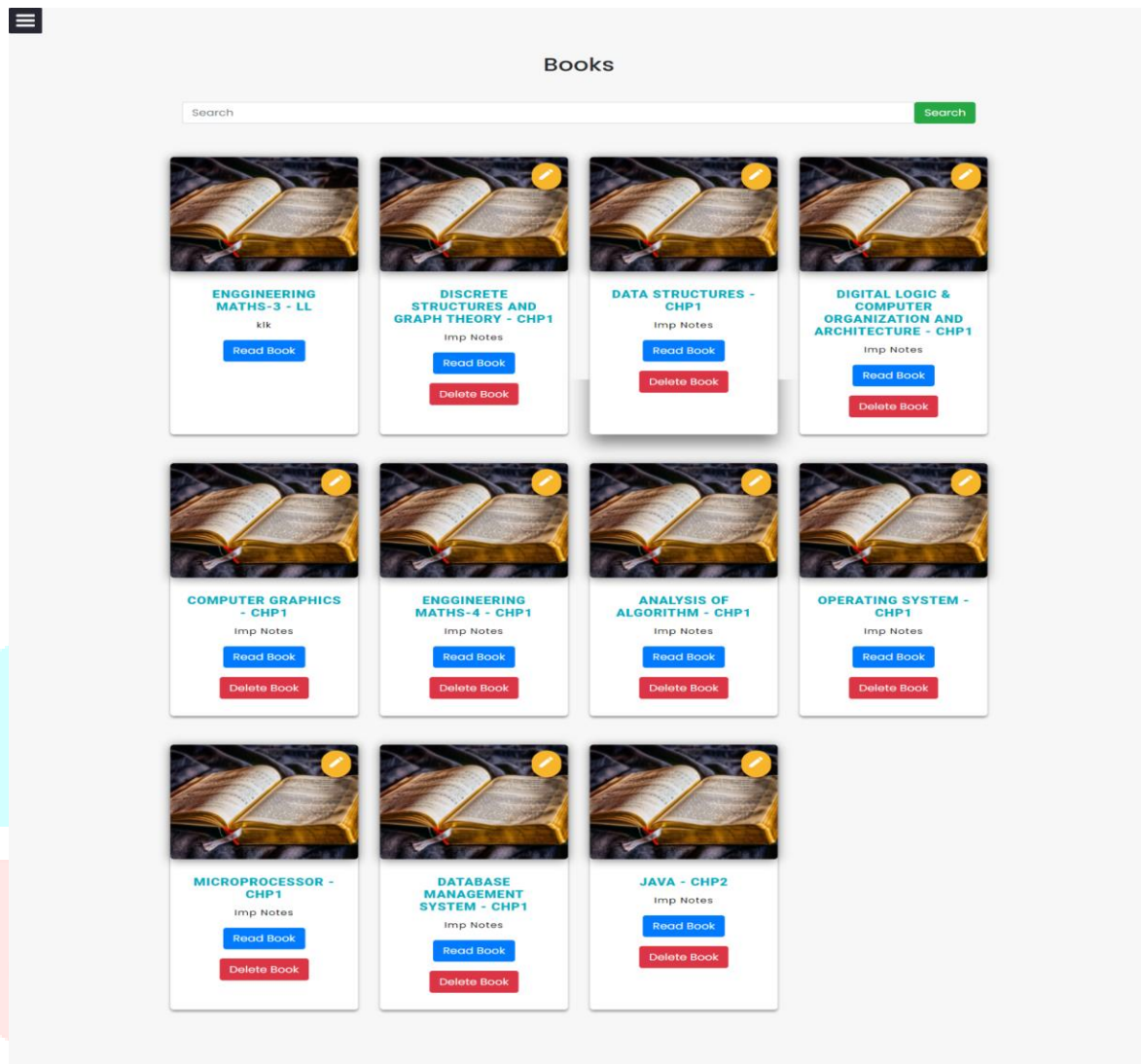


Figure 5

5. VIEW BOOKS

The View books section is available to view by all users. The user can view the books with arrows available on the side to go to the next or previous page. When the user clicks on the hamburger icon in the top left corner of the screen, they get to view the sidenav. The sidenav will generate related links to the notes once the user starts to navigate through the notes.

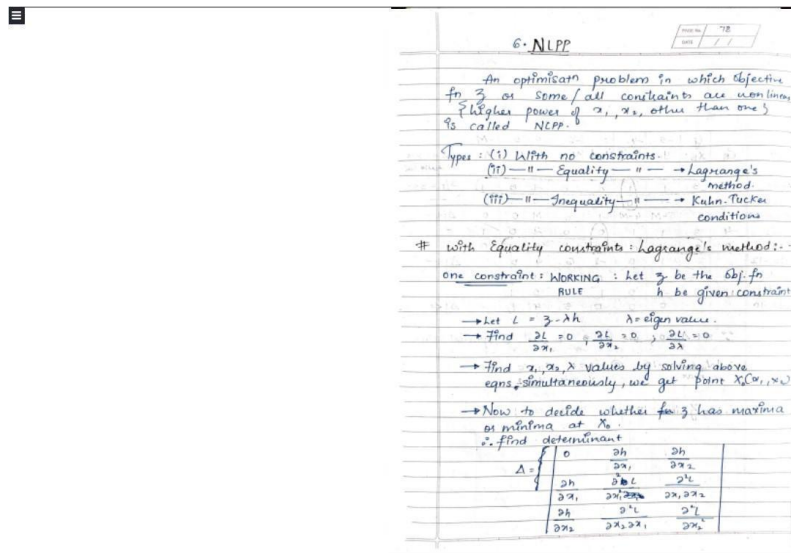


Figure 6 (a)

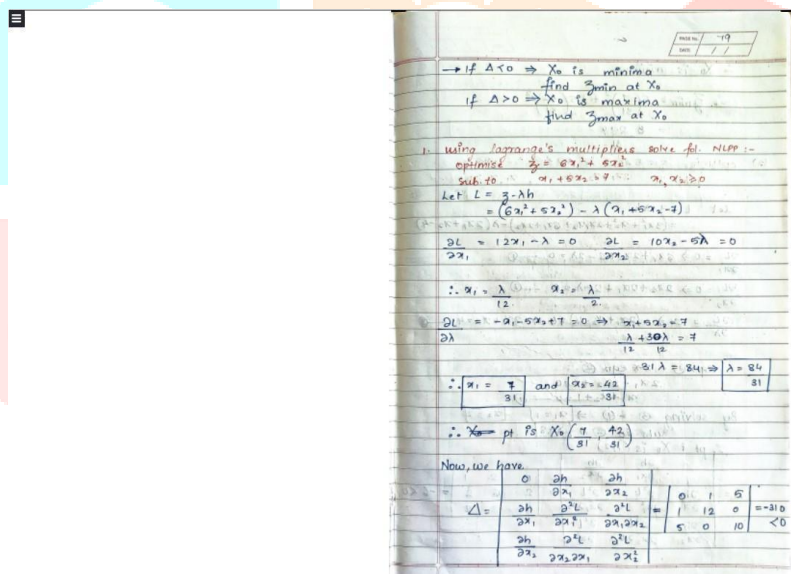


Figure 6 (b)

V. CONCLUSION

Whenever a user comes across this website, he/she has to sign in, to validate whether he/she is a valued member of the community to access the study material, and to upload the study material the faculty and students have a special email-ID as given by their college which is used to authenticate their identity.

The display page for students, faculty, and admins is designed differently. Once the user authenticates their identity as a faculty or as a student or as an admin, the appearance of the main page would be different for them.

If the user is a faculty member, they get the option to add books, edit the uploaded books, view them as well as delete the books. Faculty can upload notes in the image as well as in PDF format. Earlier the faculties were able to upload notes for only one semester, but in the new update, they can upload notes for all years (FE, SE, TE, BE). When faculties add books in image format, the images would be uploaded in the sequence in which they were selected and would be compressed when uploaded.

An admin dashboard has also been added, which is accessible by only the four of us who have contributed to the making of the project. The admin has access to all the options on the website. Admin dashboard would be useful to add new faculty members so that they can upload notes which would be accessible to the students. The students are allowed only to view the books. Now everyone can watch through a plethora of the notes uploaded by other faculty members easily from laptop or PC, whichever suits best.

As the site is very lightweight and compatible with all devices and all platforms, it can be embedded into other

websites. Also, the data gathered from the students can be helpful for the faculty in improving their notes and also giving the students a better experience next semester.

VI. REFERENCES

1. Abdul Majeed and Ibtisam Rauf: "MVC Architecture:A detailed insight to the Modern Web Applications Development Paper" Published on:September 26, 2018, (<https://drive.google.com/file/d/1DFRgbc57h8phU6no-TdAUz4AHrog6ZoH/view?usp=sharing>)
2. Google, <https://www.google.co.in/>
3. Wikipedia, <https://en.wikipedia.org/wiki/Wiki>
4. Medium, <https://medium.com/>
5. Stack Overflow, <https://stackoverflow.com/>
6. Dev.to, <https://dev.to/>
7. GeeksForGeeks, <https://www.geeksforgeeks.org/>
8. Javatpoint, <https://www.javatpoint.com/>
9. Tutorialspoint, <https://www.tutorialspoint.com/>
10. MUquestionpaper, <https://muquestionpapers.com/notes/be>

