MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Thrift Books Online Platform

Submitted By:

Subhashi Jayant

0901CS191126

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Thrift Books Online Platform

A project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Subhashi Jayant

0901CS191126

Faculty Mentor:

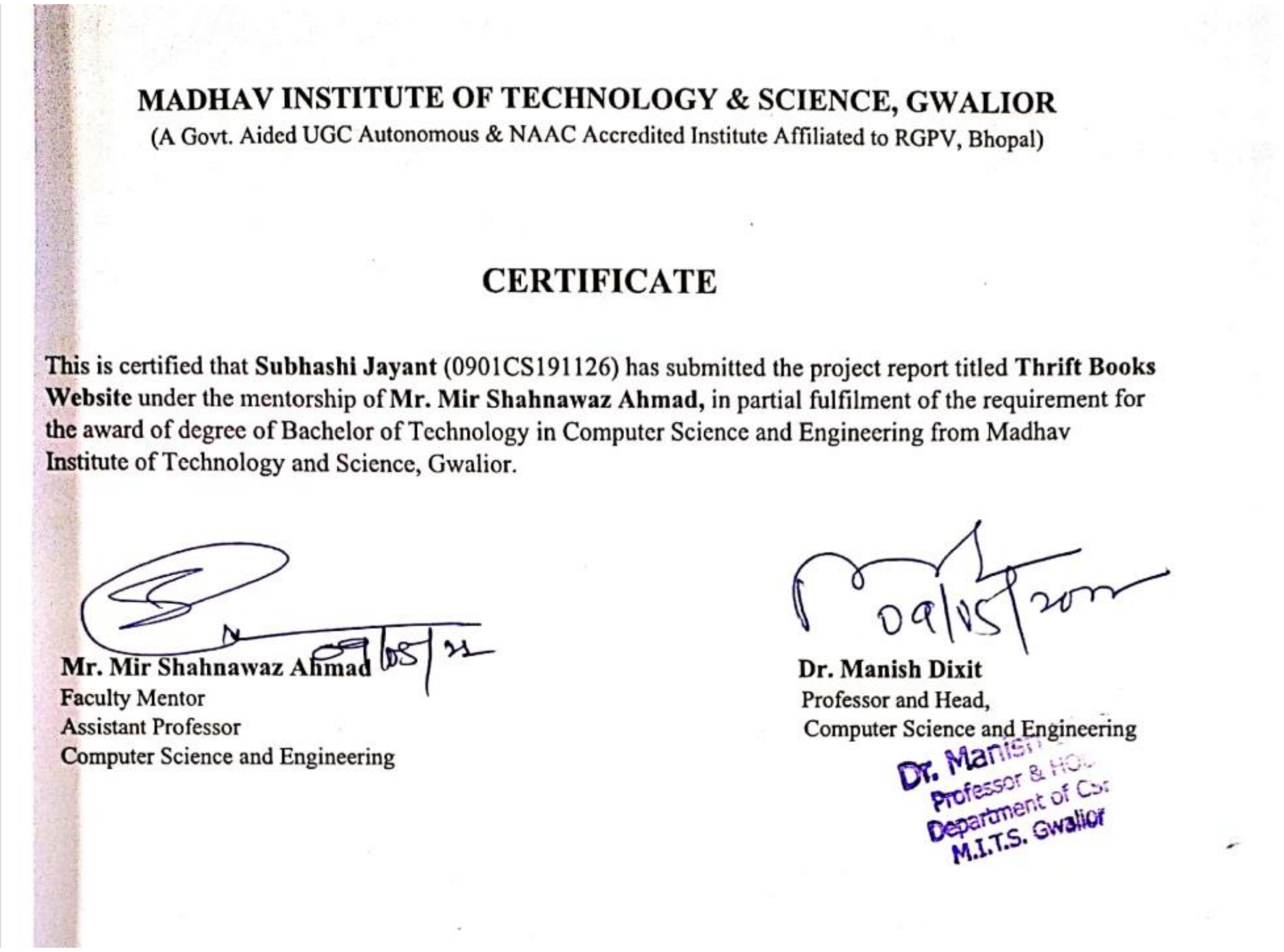
Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022



MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Subhashi Jayant** (0901CS191126) has submitted the project report titled **Thrift Books Website** under the mentorship of **Mr. Mir Shahnawaz Ahmad,** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

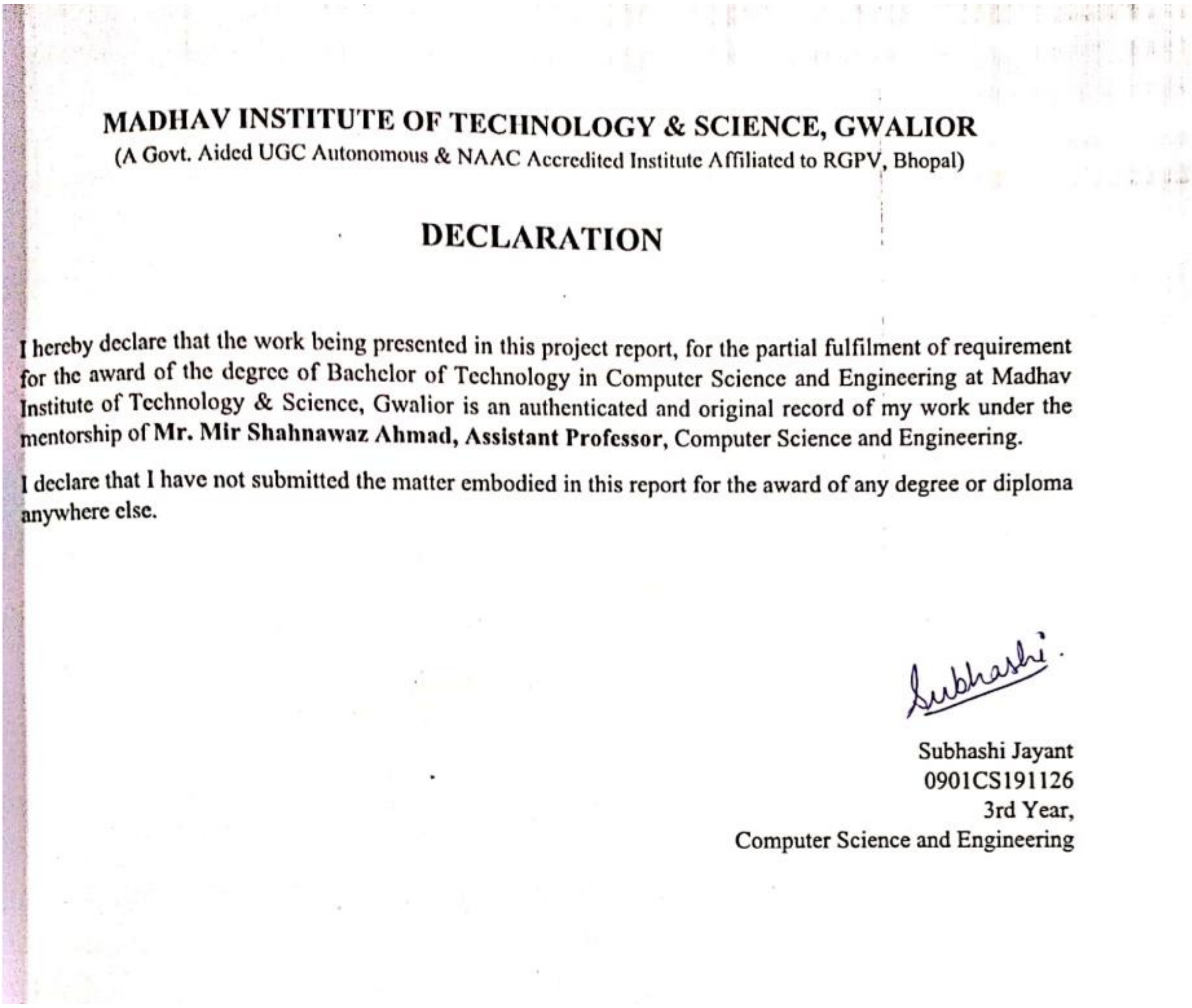
09/05/22

Mr. Mir Shahnawaz Ahmad
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manis
Professor & HO
Department of Cs
M.I.T.S. Gwalior



MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

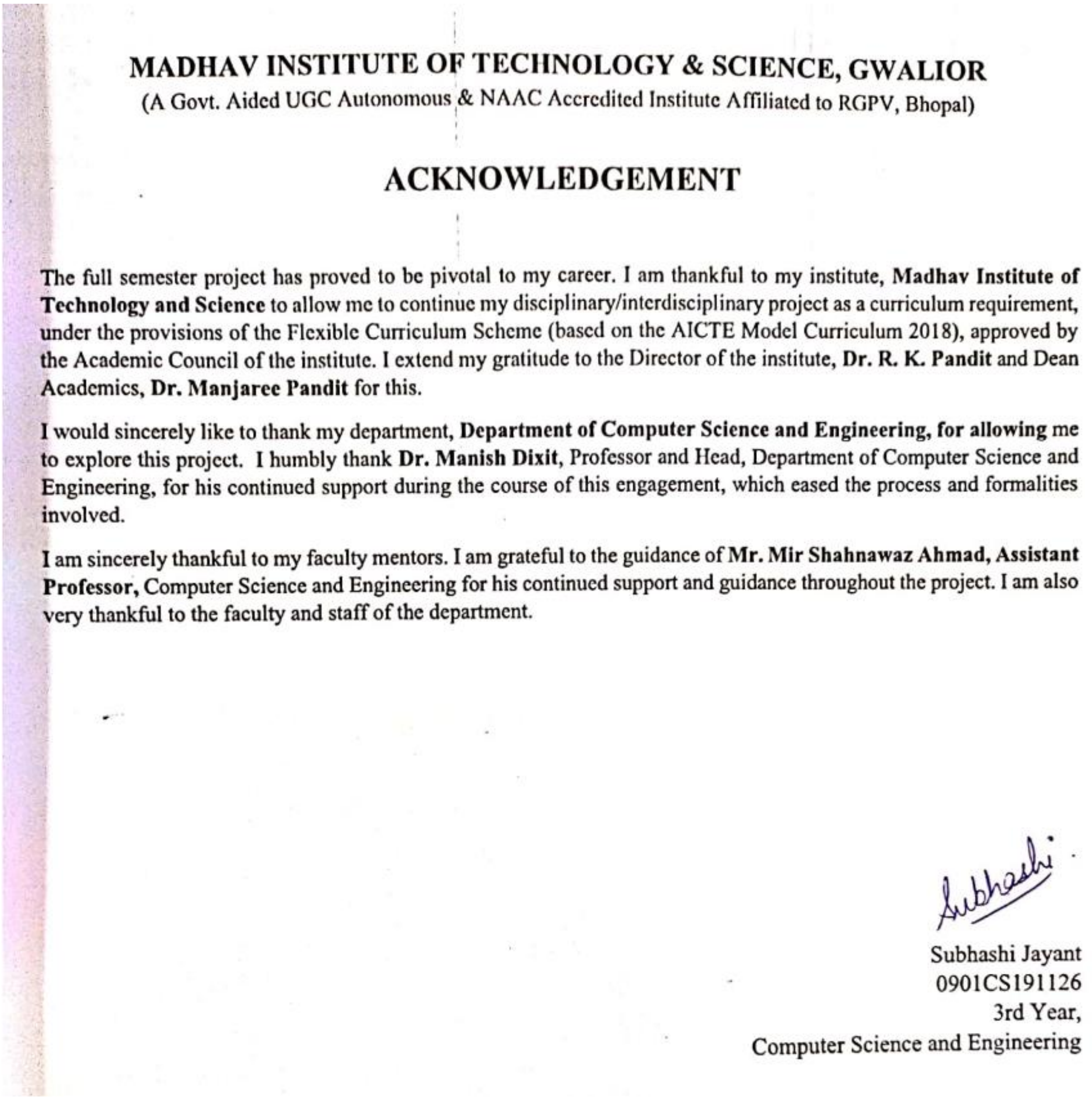
DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Mr. Mir Shahnawaz Ahmad, Assistant Professor**, Computer Science and Engineering.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Subhashi.

Subhashi Jayant
0901CS191126
3rd Year,
Computer Science and Engineering



MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Mr. Mir Shahnawaz Ahmad, Assistant Professor,** Computer Science and Engineering for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Subhashi.

Subhashi Jayant
0901CS191126
3rd Year,
Computer Science and Engineering

ABSTRACT

Buying a book is as pleasant as reading the book itself. We come across many books that catch our eye, but we end up choosing to buy only a few due to certain factors. We have several ways to read a book like an e-book, borrow a book, or take it from a library and return it after the read, but owning the book we loved reading is a different level of joy. Sometimes, we end up spending a lot of money on books, and yet we have not crossed out many books from our reading list. One answer to this problem is thrifting books. This project report discusses the development of the website, Thrift Books which lets the users to explore the collection of books and buy them at low prices. Apart from buying, the website also has the feature to let users sell their used books. The users just need to sign in and go to the My Store page and upload the details of their book. The front-end of the website is written in HTML [pug template], CSS, JavaScript and the back-end in node.js, express.js and for storage MongoDB and internal storage are used.

Keywords: Web Development, Thrift Books, Buy and Sell

सार

किताब खरीदना उतना ही सुखद है जितना कि खुद किताब पढ़ना। हमारे सामने ऐसी बहुत सी किताबें आती हैं, जो हमारी नज़रों में आती हैं, लेकिन कुछ कारणों से हम कुछ ही किताबें खरीद सकते हैं। हमारे पास किताब पढ़ने के कई तरीके हैं, जैसी ई-बुक किताब उधार लेने, या लाइब्रेरी से लेने और पढ़ने के बाद इसे वापस करने, लेकिन जिस किताब को हम पढ़ना पसंद करते हैं, उसका मालिक होना एक अलग स्तर की खुशी है। कभी-कभी, हम किताबों पर बहुत पैसा खर्च कर देते हैं, और फिर भी हमने अपनी पठन सूची से कई पुस्तकों को पार नहीं किया है। इस समस्या का एक उत्तर है मितव्ययी पुस्तकें। यह परियोजना रिपोर्ट वेबसाइट, थ्रिफ्ट बुक्स के विकास पर चर्चा करती है जो उपयोगकर्ताओं को पुस्तकों के संग्रह का पता लगाने और उन्हें कम कीमतों पर खरीदने की सुविधा देती है। खरीदने के अलावा, वेबसाइट में यह सुविधा भी है कि उपयोगकर्ता अपनी पुरानी किताबों को बेच सकते हैं। उपयोगकर्ताओं को बस साइन इन करने और माई स्टोर पेज पर जाने और अपनी पुस्तक का विवरण अपलोड करने की आवश्यकता है। वेबसाइट के फ्रंट-एंड को HTML [पग टेम्प्लेट] CSS, जावास्क्रिप्ट में लिखा गया है, और बैक-एंड में नोड.जेएस, एक्सप्रेस.जेएस और स्टोरेज के लिए मोंगोडीबी और इंटरनल स्टोरेज का उपयोग किया जाता है।

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: INTRODUCTION

1.1 Project Overview:

Buying new books can be expensive and finding a place to keep the stack of books that you no longer need is another tough task. This is where our website Thrift Books come to rescue. We provide buyers the best deals on used books. Be it your academic books or comic books, we'll introduce you to sellers who have the best deal for you. And if you are someone who has books to sell, all you need to do is upload the details of book on our website and provide buyers an attractive price and just wait and receive your first order and earn money with ease.

Apart from this, by thrifting books, you are reducing the number of new prints, eventually reducing paper usage and helping the environment.

1.2 Objectives and Scope:

- The objective of this website is to make thrifting books easier
- Let user buy used books at best prices
- Sell used books
- Make book purchases pocket and environment friendly

1.3 Project Features:

- Browse through the collection of books on the website
- Buy the books as per your likings
- Sign up and upload your used books for sale
- Search book by title
- Filter the books by category
- See similar products

1.4 Feasibility:

1.4.1 Technical feasibility: This focuses on the project's technical needs and determines how a company can meet those needs. For example, a project may require tools or software that a company doesn't currently have, and the study can help them determine whether the project merits the investment.

- The project requires cloud storage which we, currently, don't have access to.

1.4.2 Economic feasibility: This refers to cost and provides detailed information about project spending, expected revenue, projected profits and the company's return on investment. It outlines the financial benefits of the project to determine its worth.

- Cloud Storage and delivering services are the two economic constraints this project has.

1.4.3 Legal feasibility: This addresses the legal requirements of the project, such as permits or licenses. The feasibility study defines legal requirements and confirms whether compliance with legal requirements will benefit the company overall.

- Since, we're dealing with pre-owned paper-pack/ hard-cover books and not e-books, we don't need any additional copyright. Hence, the project is legally feasible.

1.4.4 Operational: Operational aspects of the feasibility study include references to how the organization will adapt to the project and how the project serves the organization's goals and mission. For example, a feasibility study might that determine a project will require additional personnel, so the company may launch a hiring initiative.

- The project is operationally feasible as it can be implemented anywhere with internet connection and system that meets the below mentioned requirements.

1.5 System Requirement:

The system requirements to build Thrift Books web application are given below.

1.5.1 Windows-Based Requirements:

Computers running Microsoft Windows must meet the following minimum Hardware and Software requirements:

- Microsoft Windows 7/8/10/11 (32- or 64- bit)
- GB RAM minimum, 8 GB RAM recommended
- GB of available disk space minimum
- 1280 * 800 minimum screen resolution

1.5.2 Software requirements:

- VS Code
- MongoDB
- MongoDB Compass

1.5.3 Hardware Requirements

- Laptop / Computer

Chapter 2: LITERATURE REVIEW

2.1 Template Engine:

A template engine enables us to use static template files in our application. At runtime, the template engine replaces variables in a template file with actual values, and transforms the template into an HTML file sent to the client. This approach makes it easier to design an HTML page.

Some popular template engines that work with Express are Pug, Handlebars, and EJS. The Express application generator uses Jade as its default, but it also supports several others.

To render template files, set the following application setting properties, set in app.js in the default app created by the generator:

- views, the directory where the template files are located. Eg: app.set('views', './views'). This defaults to the views directory in the application root directory.
- view engine, the template engine to use. For example, to use the Pug template engine: app.set('view engine', 'pug').

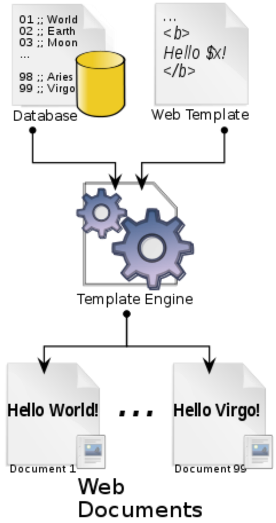


Fig 2.1 Template Engine

2.1.1 Pug Template

A template engine is a program which is responsible for compiling a template (that can be written using any one of a number of languages) into HTML. The template engine will normally receive data from an external source, which it will inject into the template it’s compiling.

Pug is a template engine for Node and for the browser. It compiles to HTML and has a simplified syntax, which can make you more productive and your code more readable. Pug makes it easy both to write reusable HTML, as well as to render data pulled from a database.

2.2 Front-end Development:

The part of a website that the user interacts with directly is termed the front end. It is also referred to as the ‘client side’ of the application. It includes everything that users experience directly: text colours and styles, images, graphs and tables, buttons, colours, and navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development. The structure, design, behaviour, and content of everything seen on browser screens when websites, web applications, or mobile apps are opened up, is implemented by front End developers. Responsiveness and performance are two main objectives of the Front End. The developer must ensure that the site is responsive i.e., it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

2.2.1 HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

2.2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

2.2.3 JavaScript

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

2.3 Back-end Development:

Backend is the server-side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

2.3.1 Nodejs

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications.

2.3.2 Express

Express.js, or simply Express, is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.[3] It has been called the de facto standard server framework for Node.js. Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library.

2.4 Storage:

There are many options where the user data can be stored:

- MongoDB, MySQL, oracle, etc
- Cloud Storage
- Internal storage

We may use any of the option depending on the need of the application as well as critically of the data.

2.4.1 MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

Main features: Ad-hoc queries, Indexing, Replication, Load balancing, File storage, Aggregation, Server-side JavaScript execution, Capped collections, Transactions

2.4.2 Internal Storage

The OS allows the data directly stored on the internal storage of the devices. The data that is stored in the file is private to the application and the other application the is present in the devices cannot access the data stored. The user(owner) of the device cannot also access the data of the application. When the application is uninstalled, the data will we removed automatically from the internal storage of the device.

Chapter 3: PRELIMINARY DESIGN

3.1 Software Development Life Cycle:

3.1.1 Rapid application development:

Rapid application development is an agile software development approach that focuses more on ongoing software projects and user feedback and less on following a strict plan. It emphasizes rapid prototyping over costly planning. The RAD model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

Since the time was limited and the planning and development happened parallelly, Rapid Application Development was chosen.

3.2 Tools & Technologies:

3.2.1 Visual Studio Code: Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++ and Fortran. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

3.2.2 MongoDB Compass: Compass is the GUI for MongoDB. Compass is an interactive tool for querying, optimizing, and analysing MongoDB data. It provides key insights, drag and drop to build pipelines, and more.

3.2.3 Web Development: Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e., websites.

Web Development can be classified into two ways:

- Frontend Development
- Backend Development

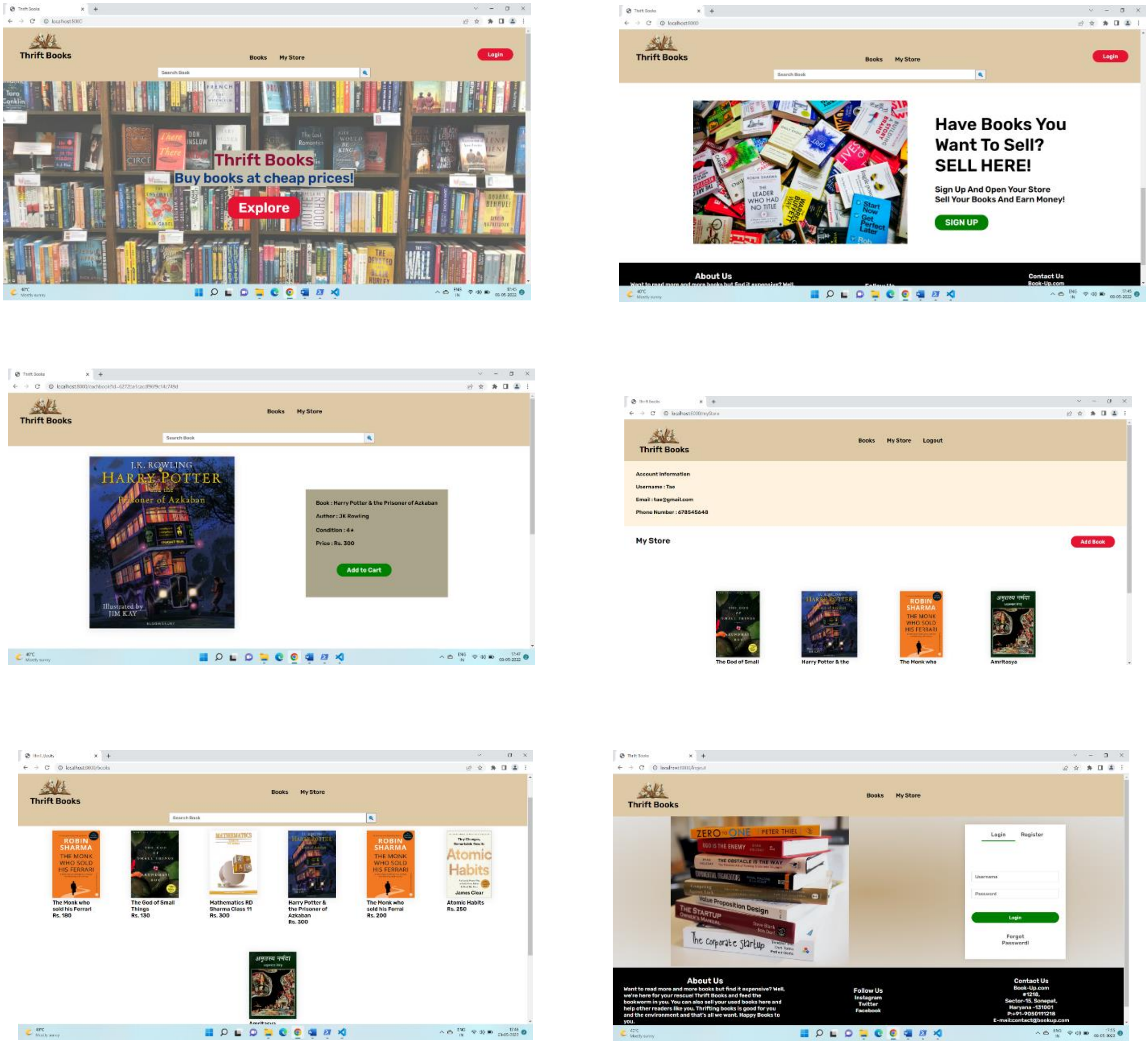
3.2.3.1 Frontend Development: The part of a website that the user interacts directly is termed as front end. It is also referred to as the ‘client side’ of the application.

3.2.3.2 Backend Development: Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

Chapter 4: FINAL ANALYSIS AND DESIGN

4.1 Results:

- The result was a multipage responsive website. On hitting the explore button, it leads us to the all books page.
- Each book is made as a link and on clicking the book, we go to the specific book page. This is all done using the pug template. The book id is send over as query parameter which is extracted and then the book page is loaded with the details of the book with the extracted id. And the books of same genre are displayed below.
- Upon login the user can access the My Store page and check their details as well as add books for sale.



Book : Harry Potter & the Prisoner of Azkaban
Author : JK Rowling
Condition : 4★
Price : Rs. 300

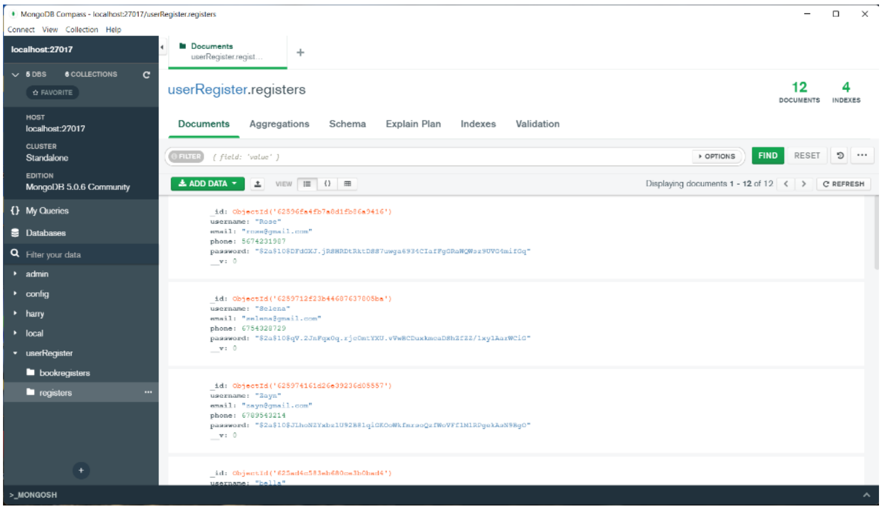
Account Information
Username : Tae
Email : tae@gmail.com
Phone Number : 678545648

The Monk who sold his Ferrari Rs. 180
The God of Small Things Rs. 130
Mathematics RD Sharma Class 11 Rs. 300
Harry Potter & the Prisoner of Azkaban Rs. 300
The Monk who sold his Ferrai Rs. 200
Atomic Habits Rs. 250

Fig 4.1.1 User Interface

4.2 Result Analysis:

- The user interface of the website is user friendly and easy to navigate.
- The website is responsive.
- The user can add book by filling out a form asking for book image, book name, author name, condition of book and the price.
- Currently, the database does not have many items to display and limited features are available.


MongoDB Compass - localhost:27017/userRegister.registers
Connect View Collection Help
localhost:27017
5 DBS 6 COLLECTIONS
FAVORITE
HOST localhost:27017
CLUSTER Standalone
EDITION MongoDB 5.0.6 Community
My Queries
Databases
Filter your data
admin
config
harry
local
userRegister
bookregisters
registers
Documents userRegister.regist...
userRegister.registers
12 DOCUMENTS
4 INDEXES
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS FIND RESET
ADD DATA VIEW
Displaying documents 1 - 12 of 12
REFRESH
>_MONGOSH

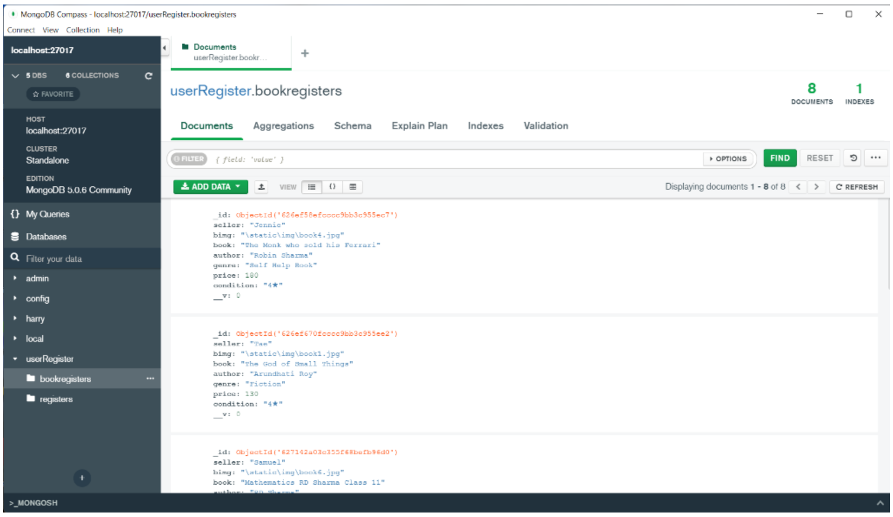
```
_id: ObjectId('62596fa4fb7a8d1fb86a9416')
username: "Rose"
email: "rose@gmail.com"
phone: 5674231987
password: "$2a$10$DFdGXJ.jR8HRDtRktD887uwga6934CIafFgGRaWQWsz9UVG4mifGq"
__v: 0

_id: ObjectId('6259712f23b44607637805ba')
username: "Selena"
email: "selena@gmail.com"
phone: 6754328729
password: "$2a$10$qV.2JnFqx0q.rjcOmtYXU.vVwBCDuxkmcaD8hZfZZ/1xy1AazWCiC"
__v: 0

_id: ObjectId('625974161d26e39236d05557')
username: "Zayn"
email: "zayn@gmail.com"
phone: 6789543214
password: "$2a$10$JLhoN2Yxbz1U92B81qiGKOoWkfmrsoQzfWoVFf1M1RPgekAaN9BgO"
__v: 0

_id: ObjectId('625ad4c583eb680ce3b0bad4')
username: "bella"
```

Fig 4.2.1 User Database


MongoDB Compass - localhost:27017/userRegister.bookregisters
Connect View Collection Help
localhost:27017
5 DBS 6 COLLECTIONS
FAVORITE
HOST localhost:27017
CLUSTER Standalone
EDITION MongoDB 5.0.6 Community
My Queries
Databases
Filter your data
admin
config
harry
local
userRegister
bookregisters
registers
Documents userRegister.bookr...
userRegister.bookregisters
8 DOCUMENTS
1 INDEXES
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS FIND RESET
ADD DATA VIEW
Displaying documents 1 - 8 of 8
REFRESH
>_MONGOSH

```
_id: ObjectId('626ef58efcccc9bb3c955ec7')
seller: "Jennie"
bimg: "\static\img\book4.jpg"
book: "The Monk who sold his Ferrari"
author: "Robin Sharma"
genre: "Self Help Book"
price: 180
condition: "4★"
__v: 0

_id: ObjectId('626ef670fcccc9bb3c955ee2')
seller: "Tae"
bimg: "\static\img\book1.jpg"
book: "The God of Small Things"
author: "Arundhati Roy"
genre: "Fiction"
price: 130
condition: "4★"
__v: 0

_id: ObjectId('627142a03c355f68befb96d0')
seller: "Samuel"
bimg: "\static\img\book6.jpg"
book: "Mathematics RD Sharma Class 11"
author: "RD Sharma"
```

Fig 4.2.2 Book Database

4.3 Problems Faced:

- This web application requires image input by the user, which are recommended to be stored in the cloud storage but since it is paid, internal storage was used.
- Most websites show shopping cart without having to login but since I was using various accounts to test the website, it wasn't possible implement cart feature.

4.4 Limitations:

- Since the website uses high resolution images, the loading time is more.
- Currently, there is no solution if the buyer wants to return the product.
- The database is highly user dependent.

Chapter 5: CONCLUSION

5.1 Conclusion:

With this project, our purpose was to provide users access to used books at low prices and to provide them a platform to sell their used books that are no longer needed. The website has been developed as an Express application. We made use of PUG template engine to use inline JavaScript and render dynamic content. In order to make the website secure, the passwords have been hashed using bcrypt. The users just need to sign in by providing their general information and can get access to all of the features of the web application.

Thrift books Website will not only help readers to get best deals but it is also environment friendly as we're reducing paper usage by not printing new books.

5.2 Future Scope:

The website can be made better and more user friendly by:

- Letting user edit their uploaded books.
- The search feature which is currently limited to search by title can be upgraded.
- Filtering books by author name, price range etc. can be added.
- The website needs a fully functioning cart page.

REFERENCES

- Using template engines with Express – Expressjs.com
- Pug HTML Template Engine: A Beginner's Guide – SitePoint
- Web Development Complete Tutorial – codewithharry.com
- MongoDB Tutorial – thapatechnical.com