CHAPTER 1

INTRODUCTION

Proposed Book Finder is an simple website that will allow users to search for books by title, author name . In the present scenario, we are dealing with the manual searching of book from thousands of other books present in Google Book API. With the help of this application we are able to find a place where we can easily find the book with the help of keywords. This application takes in a user input and searches the Google Books API with the user input and gets a list of published books based on the users search query To get the information of the particular book user can click upon the book from the list and then will be taken to the new tab where description and other information related to the book will be available. Users can also add the book to the favourites.

1.1 OBJECTIVE

The main objective of our Book Finder Website is to provide an easy-to-use for students ,teens ,and educators in order to discover diverse books of different genres.

This website will give readers new ways to show their love for books, review them online, and, of course, discover new books to read.

1.2 PROBLEM STATEMENT

Book finder process is well organized online buying of books. This system is well developed in various resources, for example Amazon site deals more about e-booking concept. This process has various issues in the basics of maintenance of database and updating in sites, and virus problem in pdf books, so we have many issues in this process. The process of book finder is fully based on online, and the process for this mainly interaction between buyer and seller, buyer who enter the site for purchase of book will use search engine for book to purchase, the search engine will mainly focused on the database process, it used to search book for the buyer who mentioned the book name, author name, edition, publication details in the site, so that the search engine will show many books. There will be a payment option.

1.3 PURPOSE

If the entire process of 'Issue of Books or Magazines' is done in a manual manner then it would take several months for the books or magazines to reach the applicant. Considering the fact that the number of students for Book Bank is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. The system has been carefully verified and validated in order to satisfy it.

1.4 SCOPE

The System provides an online interface to the user where they can fill in their personal details and submit the necessary documents (may be by scanning). The authority concerned with the issue of books can use this system to reduce his workload.

1.5 MOTIVATION

As we all have been facing a crucial lockdown periodic due to the pandemic, we have realized the importance of virtual learning and how important it is for us to have our resources online. And along with this we need to have a place to keep the resources for areas of our interest so we thought of developing this website and hence implementing it.

This website will be free of cost so all the students interested to learn can easily learn from it. Moreover this kind of website can be used in areas/schools where guardians /parents cannot afford to buy books. This would be an excellent effort to provide education without any boundaries to all.

CHAPTER 2

SYSTEM ANALYSIS

2.1 LITERATURE REVIEW

In today's world, there are more books available than ever before. This can make it difficult for people to find the books they are looking for. A book finder project can help to solve this problem by providing a way for people to search for books based on their interests.There is a growing body of research on book finding tools and techniques. This research has identified a number of factors that influence the effectiveness of book finding tools, including:

- The quality of the data used to train the tool
- The user interface
- The features offered by the tool
- The quality of the data used to train a book finding tool is one of the most important factors that influences its effectiveness.

A tool that is trained on a large and diverse dataset of books will be more likely to be able to find relevant books for users.The user interface of a book finding tool is also important. A tool with a clear and easy-to-use interface will be more likely to be used by people.The features offered by a book finding tool can also influence its effectiveness. A tool that offers a variety of features, such as the ability to search by author, title, genre, and keyword, will be more likely to be able to find the books that users are looking for.

2.1.1 RESEARCH QUESTIONS

The following research questions will be addressed in this literature review:

- What are the different types of book finding tools?
- What are the strengths and weaknesses of different book finding tools?
- What are the factors that influence the effectiveness of book finding tools?
- How can book finding tools be improved?
- Types of Book Finding Tools

- There are a variety of book finding tools available, ranging from simple

Simple search engines are the most basic type of book finding tool. They allow users to search for books by author, title, or keyword. Simple search engines are not very effective at finding relevant books, as they do not take into account the user's interests.Complex recommendation systems are more effective than simple search engines at finding relevant books. They use algorithms to analyze the user's past reading history, interests, and ratings to recommend books that the user is likely to enjoy.Strengths and Weaknesses of Book Finding ToolsThe strengths and weaknesses of different book finding tools vary depending on the type of tool.Simple search engines are easy to use and can be accessed from anywhere. However, they are not very effective at finding relevant books.Complex recommendation systems are more effective at finding relevant books than simple search engines. However, they can be more difficult to use and may not be as accurate as simple search engines.Factors that Influence the Effectiveness of Book Finding ToolsThe effectiveness of book finding tools is influenced by a number of factors, including:

- The quality of the data used to train the tool
- The user interface
- The features offered by the tool
- The quality of the data used to train a book finding tool is one of the most important factors that influences its effectiveness.

A tool that is trained on a large and diverse dataset of books will be more likely to be able to find relevant books for users.The user interface of a book finding tool is also important. A tool with a clear and easy-to-use interface will be more likely to be used by people.The features offered by a book finding tool can also influence its effectiveness. A tool that offers a variety of features, such as the ability to search by author, title, genre, and keyword, will be more likely to be able to find the books that users are looking for.

2.1.2 HOW TO IMPROVE BOOK FINDING TOOLS

There are a number of ways to improve book finding tools. One way is to improve the quality of the data used to train the tool. Another way is to improve the user interface of the tool. Finally, tools can be improved by adding new features, such as the ability to search by author, title, genre, and keyword. There are a number of factors that influence the effectiveness of book finding tools. The quality of the data used to train the tool, the user interface, and the features offered by the tool all play a role in determining how effective a tool is. By improving these factors, book finding tools can be made more effective and user-friendly.

2.2 EXISTING SYSTEM

It has been a part of the existing system for as long we can remember In the existing system, we can store all the record manually that require large manpower & place to store all the records this system was carried out through a manual process.

2.2.1 DRAWBACKS IN EXISTING SYSTEM

- Slowupdating & renewal of data.
- When a user requests for the a book, one has to physically check for the presence of a book in the library.
- Daily keeping a manual record of changes taking place in the library such as book being issued, book being returned etc.

2.3 PROPOSED SYSTEM

Automated To overcome the disadvantages of the existing system we proposed the GUI system. The best technology learning tool such as interactive software are used by us to provide consuming the time

2.3.1 ADVANTAGES

- The online website is time-saving and better performance than manual system

- Easy search of book in the online library.
- Avoid the manual work.

2.4 PRESENT WORK OF THE PROJECT:

For the purpose of this project, I'll use the Google Book API to have a real database for the search. In order to know more about Google Book API, click on the above link. Google Book API is intended for developers who want to write applications that can interact with the Books API. Google Books has a mission to digitize the world's book content and make it more discoverable on the Web. The Books API is a way to search and access that content, as well as to create and view personalization around that content.

2.4.1. USING THE API

The Google BookAPI End Point is :

https://www.googleapis.com/books/v1/volumes?q=$%7Bquery%7Ds&key=$%7Bapi_key%7D

The API has a lot of parameters that we can use, but we'll only need 3 of them which are :

- **q**: the search query tapped by the user in the search input.
- **maxResults** : The maximum number of results to return. The default is 10.
- **client_id** : the client ID generated in your Google Books Console account. In order to use the Google Books API, you have to create a developer account and register your app and also to generate an API-KEY.

2.4.2. CREATING OUR SEARCH SERVICE

ng generate api service

This command will create an api.service.ts file in the app folder, and register this service as provider in app.module The service simply makes a GET request to the Google Books API, and return the search result as an observable. Now that out our service is ready

to make an API call, let now connect out search input and the google book API to send value to the database.

2.4.3 THE SOLUTION APPROACH

Let’s think of an approach, after user types in a query of three characters then we can make an API Hit. In this case, only one meaningful Hit will be made instead of three. We can wait for the user to type the whole query first after that we can make an API call. But how can we do that programmatically? and also how to stop sending an empty value when we tab space without writing a word.

2.4.4 VIEW.COMPONENT.TS

```
.subscribe((query: any) =>
{
        let te = query.replace(/\s/g, "");
}
```

In the above code we create a variable and set it to be equal to the valueof what we get from our input and we use **.replace(/\s/g, "")** function to remove space and to stop our input from sending an empty value. But I know you would want to know what expression is that and why it worked. This explanation will help: It's a regular expression where the \s means "match whitespace" and the g is a flag which means "global", i.e. match all whitespace, not just the first. Let's remember that it was two problems we encountered and the above code solved just one and we are let with one.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. Face recognition access requires using specialized reader hardware.They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

RAM: 4GB

Hard Disk: 500GB

Processor: Dual Core processor

3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system do rather than how it should do it. The software requirements provide a basic for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development.

Operating System: Windows 10

Coding Language: HTML, CSS, JAVASCRIPT

Processing software: Liveserver

3.3. SOFTWARE DESCRIPTION

In a desktop website like Book Finder System, there is a scope for a large number of platforms, languages and frameworks to choose from. Before selecting from this large array of technologies, the following aspects, which are characteristic to windows based application like this one, have been kept in mind:

- Performance
- Reliability
- Scalability
- Portability
- Performance
- Time constraint
- Cost constraint

3.4 THE VARIOUS TECHNOLOGIES AVAILABLE FOR CONSIDERATION ARE AS FOLLOWS

3.4.1 OPERATING SYSTEM

- Windows 7

3.4.2 FRONT END

- HTML
- CSS

3.4.3. PROGRAMMING LANGUAGE

- JavaScript

3.4.4. TESTING SERVER

- Liveserver

3.5 HTML

HTML or HyperText Markup Language is the standard markup language used to create web pages.HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

3.6 CSS

CSS was first developed in 1997, as a way for Web developers to define the look and feel of their Web pages. It was intended to allow developers to separate content from design so that HTML could perform more of the function that it was originally based on the markup of content, without worry about the design and layout. CSS didn't gain in popularity until around 2000, when Web browsers began using more than the basic font and color aspects of CSS.Web Designers that don't use CSS for their design and development of Web sites are rapidly becoming a thing of the past. And it is arguably as important to understand CSS as it is to know HTML - and some would say it was more important to know CSS. Style sheet refers to the document itself. Style sheets have been used for document design for years. They are the technical specifications for a layout, whether print or online. Print designers use style sheets to

insure that their designs are printed exactly to specifications. A style sheet for a Web page serves the same purpose, but with the added functionality of also telling the viewing engine (the Web browser) how to render the document being viewed.

3.7 JAVASCRIPT :

JavaScript is a high-level, dynamic, interpreted programming language that is widely used in web development to create interactive and dynamic web pages. It was initially developed in the mid-1990s by Brendan Eich at Netscape, and has since become one of the most widely used programming languages on the web. JavaScript is a versatile language that can be used for a wide range of tasks, including adding interactivity and dynamic functionality to web pages, creating animations and games, building server-side applications, and more. It is designed to be easy to use and learn, with a syntax that is similar to other popular programming languages like C and Java.One of the key features of JavaScript is its ability to manipulate the Document Object Model (DOM) of a web page. The DOM is a tree-like structure that represents the content and structure of a web page, and JavaScript allows developers to add, remove, and modify elements of the DOM in real-time. This enables developers to create dynamic and responsive web pages that can react to user actions and input. JavaScript is also used for event handling, which allows developers to respond to user actions such as mouse clicks, keyboard inputs, and form submissions. This is an essential feature for creating interactive web applications that can provide a more engaging and user-friendly experience. In addition to client-side scripting, JavaScript is also used for server-side scripting through platforms like Node.js. This allows developers to use JavaScript to build scalable and efficient server-side applications that can handle a large number of concurrent users and requests. JavaScript has a large and active community of developers who contribute to a wide range of open-source libraries and frameworks that make it easier to develop complex web applications. Some popular libraries and frameworks include React, Angular, and Vue.js. Overall, JavaScript is a powerful and versatile programming language that is essential for modern web

development. Its ability to manipulate the DOM and handle events makes it an ideal choice for creating dynamic and interactive web pages, while its support for server-side scripting allows developers to build scalable and efficient web applications. With its ease of use and flexibility, JavaScript will continue to play a crucial role in web development for years to come.

3.8 WEB SERVER

Back-end Development refers to the server-side development. It focuses on databases, scripting, website architecture. It contains behind-the-scene activities that occur when performing any action on a website. It can be an account login or making a purchase . Code written by back-end developers helps browsers to communicate with database information.

3.9 LIVESERVER

LiveServer is a lightweight and easy-to-use web development tool that provides a simple way to create and test web applications. It is designed to simplify the web development process by providing a local server environment that automatically reloads pages whenever changes are made to the source code. LiveServer is a browser-based application that can be used with any web browser. It is compatible with most popular web development languages, including HTML, CSS, JavaScript, and PHP. It is easy to install and can be used with any text editor or integrated development environment (IDE). One of the key features of LiveServer is its automatic reloading functionality. Whenever changes are made to the source code of a web page or application, LiveServer automatically refreshes the page in the browser, allowing developers to see the changes in real-time. This greatly speeds up the development process, as developers no longer need to manually refresh the page every time they make a change. LiveServer also provides a range of other useful features that make it easier to develop and test web applications. For example, it includes support for custom domain names, which allows developers to use domain names other than "localhost" when testing their applications. It also provides support for HTTPS, which is essential for testing secure web

applications. Another useful feature of LiveServer is its support for multiple devices. Developers can use LiveServer to test their applications on a range of different devices, including smartphones and tablets, making it easier to ensure that their applications work properly across different platforms. LiveServer is also designed to be highly customizable. It provides a range of configuration options that allow developers to customize the way that the application works, including the ability to specify custom port numbers and URL paths. It also provides support for browser sync, which enables multiple devices to synchronize changes in real-time. Overall, LiveServer is a powerful and flexible tool that makes web development faster and easier. Its automatic reloading functionality, support for multiple devices, and customizability make it an essential tool for any web developer looking to streamline their workflow and improve their productivity.

CHAPTER 4

PROBLEM DESCRIPTION

4.1 PROBLEM STATEMENT

Book finder process is well organized online buying of books. This system is well developed in various resources, for example Amazon site deals more about e-booking concept. This process has various issues in the basics of maintenance of database and updating in sites, and virus problem in pdf books, so we have many issues in this process. The process of book finder is fully based on online, and the process for this mainly interaction between buyer and seller, buyer who enter the site for purchase of book will use search engine for book to purchase, the search engine will mainly focused on the database process, it used to search book for the buyer who mentioned the book name, author name, edition, publication details in the site, so that the search engine will show many books. There will be a payment option .

4.2 OVERVIEW OF THE PROJECT

Bookfinder is an online book search engine that utilizes the Google Books API to help users find and compare prices of books from different sellers. When a user enters a book's title, author, into Bookfinder's search bar, the website uses the Google Books API to search through millions of books and return relevant results. The user can then view the book's details, such as its cover image, publication date, and synopsis, and compare prices and availability from different online bookstores and marketplaces. Bookfinder provides a user-friendly interface that allows users to easily filter results by book format, language, and price range. The website also provides useful information about each seller, such as their shipping fees and estimated delivery time, to help users make informed buying decisions. Overall, Bookfinder is a convenient and efficient tool for book lovers to find and purchase books online from a variety of sources.

CHAPTER 5

SYSTEM DESIGN

5.1 USE-CASE DIAGRAM

The E-book use cases in our system are:

- Login
- Search book
- Download
- Payment
- Publisher
- Update

5.1.1 ACTORS INVOLVED

- User
- Visitor
- Administrator

5.1.2 ADD TO BOOK ITEM

- Each book should have following attribute
- title
- Author name

5.1.3 QUERY THE BOOK DATABASE

The product shall let Google Api query tools books detail information by their author (or) title. The search result would produce a list of books, which match the search parameters.

5.1.4 CHECK OUT A BOOK

User can check out can be initialized from a previous search operation where user has selected a set of books.

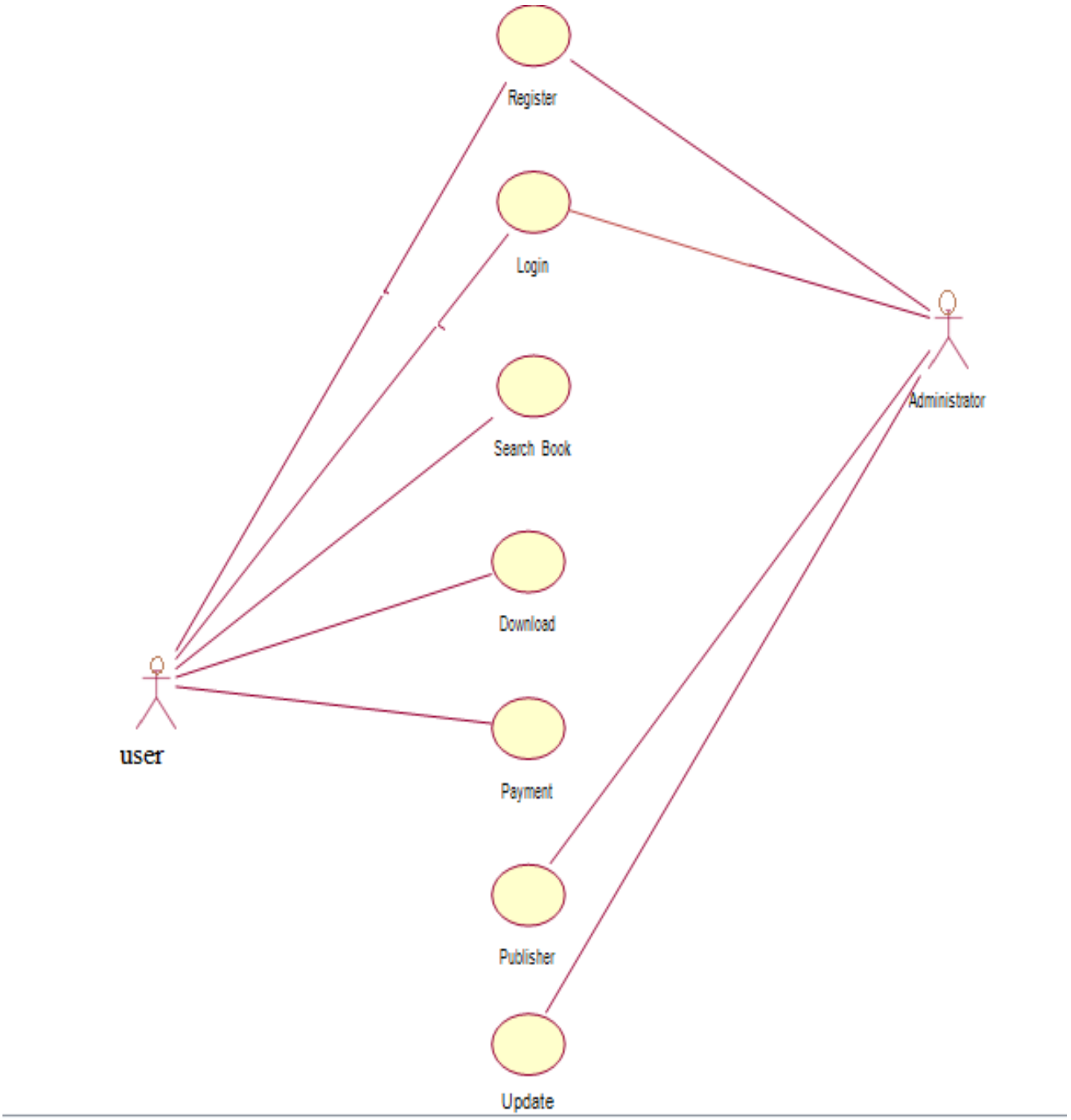

Figure 5.1.1: Use-Case Diagram

5.2 ACTIVTY DIAGRAM

The activity diagram shows the activity of the process here first login is done when the user is valid then the welcome page appears .Here fork is used where two transaction line may be got search book and online reading .search book can be used to search book and online reading can allow user to learn online and when any of these two process is selected a join is used where download occurs, in this download of book is done then finally cost of book is paid online.

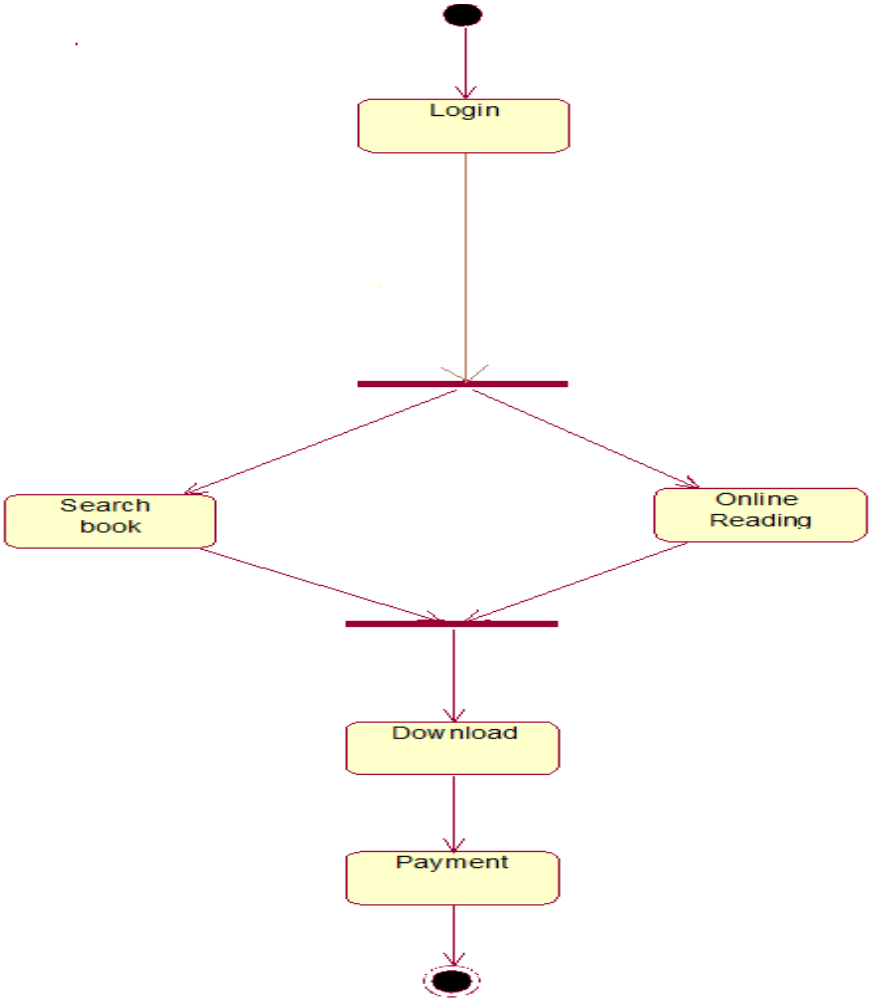

Figure 5.2.1: Activity Diagram

5.3 INTERACTION DIAGRAM

A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

- An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another, in which the "form" object is requesting an operation be performed by the "to" object. The "to" object performs the operation using a method that the class contains.
- It is also represented by the order in which things occur and how the objects in the system send message to one another

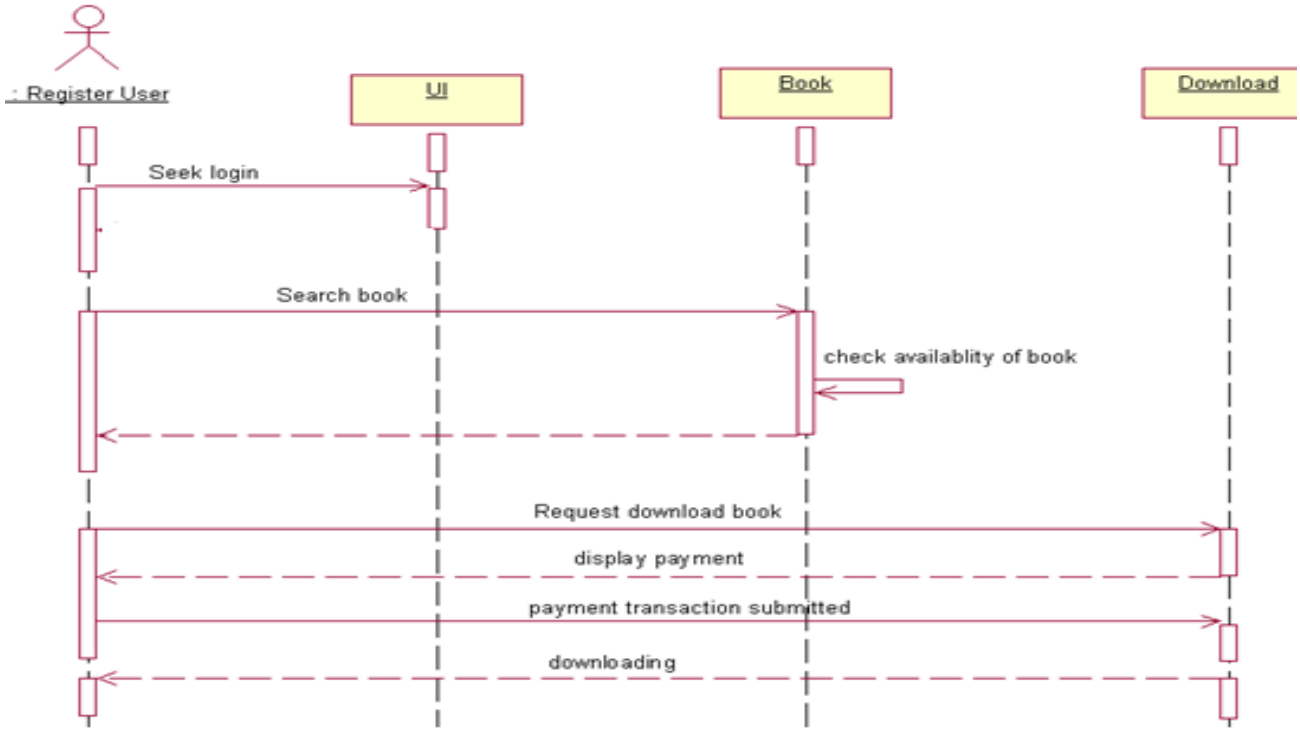

Figure 5.3.1: Sequence Diagram For User

5.4. STATE CHART DIAGRAM

The diagrams show first login to the system and view the books and search for required book is done and then required book is downloaded and amount paid in online.Finally Exit from the system.

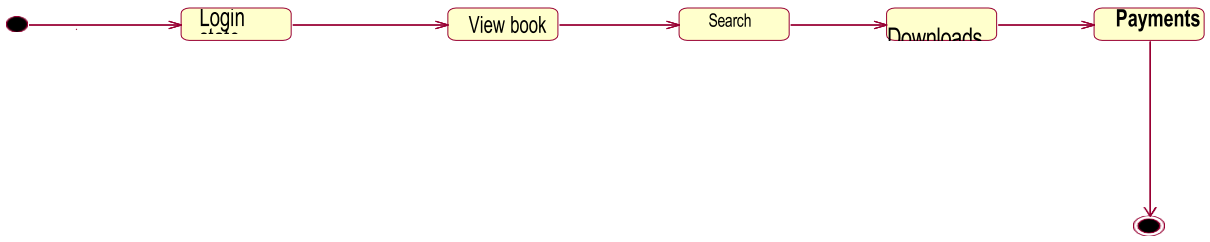

Figure 5.4.1: State Chart Diagram

5.5. DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

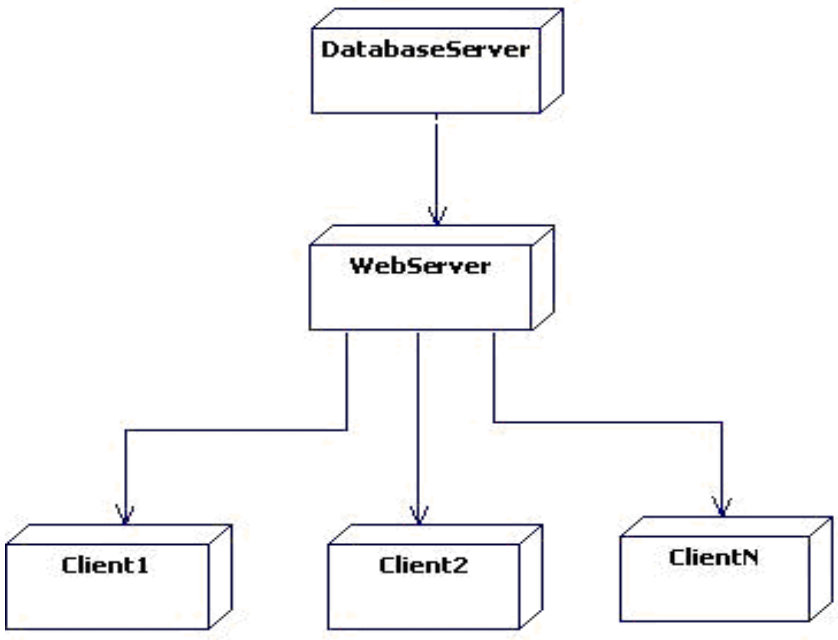

Figure 5.5.1 : Deployment Diagram for Physical Component of a System

5.6 INPUT DESIGN

Very careful attention had to be given to input design, which is a major part of the overall system design. In order to make the data entry as easy, logical and error free as possible, specific standards had been followed. Validation checks, provided in the system prevented the user in entering incorrect, erroneous data. This made sure that, only valid data had been available for data processing. If valid data was entered, then meaningful error messages had been prompted to enter correct data. The interactive screen formats facilitate the entry of valid data.

5.7 VALIDATIONS

Some fields are having only number, as an I/P. For this key ASCII is checked. If they entered characters, it would display the message to enter number only. Exchange rates field will be validated for number and dot symbols.

5.8 INPUT DESIGN OBJECTIVES

The numbers of clear objectives of input design are,

- To produce a cost effective method of input
- To achieve the highest possible level of accuracy
- To ensure that the input is acceptable to and understand by the user staff

5.9 OUTPUT DESIGN

Output, as you probably know, generally refers to the results and information that are generated by the system. For many end-users, output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the website. Most end users will not actually operate the information system or enter data through workstations, but they will use the output from the system.

When designing output, systems analysts must accomplish the following.

- Determine what information to present

- Decide whether to display, print, or “speak” the information and select the output medium.
- Arrange the presentation of information in an acceptable format.
- Decide how to distribute the output to intended recipients.
- That alignment of information on a display or printed document is termed as layout.

Accomplishing the general activities listed above will require specific decisions, such as whether to use preprinted forms when preparing reports and documents, how many lines to plan on a printed page, or whether to use graphics and color. The output design is specified on layout performs, sheets that describe the location characteristics, and format of the column headings and pagination. As we indicated at the beginning of this discussion, these elements are analogous to an architect’s blue print that shows the location of the each component.

CHAPTER 6

SYSTEM TESTING

6.1 TESTING & EXPLANATION

The testing phase is an important part of software development. It is the puterized system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

6.1.2 SOFTWARE TESTING IS CARRIED OUT IN THREE STEPS

- The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is the important and major part of the project. So errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So unit testing is conducted to individual modules.

- The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. This is due to poor interfacing, which may results in data being lost across an interface. A module can have inadvertent, adverse effect on any other or on the global data structures, causing serious problems.

- The final step involves validation and testing which determines which the software functions as the user expected. Here also some modifications were. In the completion of the project it is satisfied fully by the end user.

6.2. MAINTENANCE AND ENVIRONMENT

AS the number of computer based systems, grieve libraries of computer software began to expand. In house developed projects produced tones of thousand soft program source statements. Software products purchased from the outside added hundreds of thousands of new statements. A dark cloud appeared on the horizon. All of these programs, all of those source statements-had to be corrected when false were detected, modified as user requirements changed, or adapted to new hardware that was purchased. These activities were collectively called software Maintenance.

The maintenance phase focuses on change that is associated with error correction, adaptations required as the software's environment evolves, and changes due to enhancements brought about by changing customer requirements. Four types of changes are encountered during the maintenance phase.

- Correction
- Adaptation
- Enhancement
- Prevention

6.2.1 CORRECTION

Even with the best quality assurance activities is lightly that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects. Maintenance is a set of software Engineering activities that occur after software has been delivered to the customer and put into operation. Software configuration management is a set of tracking and control activities that began when a software project begins and terminates only when the software is taken out of the operation.

We may define maintenance by describing four activities that are undertaken after a program is released for use:

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance or Enhancement
- Preventive Maintenance or reengineering

Only about 20 percent of all maintenance work are spent "fixing mistakes". The remaining 80 percent are spent adapting existing systems to changes in their external environment, making enhancements requested by users, and reengineering an application for use.

6.2.2 ADAPTATION

Over time, the original environment (E>G., CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate change to its external environment.

6.2.3 ENHANCEMENT

As software is used, the customer/user will recognize additional functions that will provide benefit. Perceptive maintenance extends the software beyond its original function requirements.

6.2.4 PREVENTION

Computer software deteriorates due to change, and because of this, preventive maintenance, often called software reengineering, must be conducted to enable the software to serve the needs of its end users. In essence, preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted, and enhanced. Software configuration management (SCM) is an umbrella activity that is applied throughout the software process.

6.3 TESTING OF BOOK FINDER SYSTEM

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding. System Testing is an important phase. Testing represents an interesting anomaly for the software.Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing. A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

6.3.1 TESTING OBJECTIVES

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

6.4 TESTING PRINCIPLES

- All tests should be traceable to end user requirements
- Tests should be planned long before testing begins
- Testing should begin on a small scale and progress towards testing in large
- Exhaustive testing is not possible
- To be most effective testing should be conducted by a independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are:

- White box testing.
- Black box testing.

6.4.1 WHITE-BOX TESTING

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

6.4.2 BLACK-BOX TESTING

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

6.4.3 TESTING STRATEGIES

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

6.4.4 TESTING FUNDAMENTALS

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

6.4.5. TESTING INFORMATION FLOW

Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code. Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied

and debugging commences.

6.4.6 UNIT TESTING

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules. These tests were carried out during the programming stage itself. All units of ViennaSQL were successfully tested.

6.4.7 INTEGRATION TESTING

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

6.4.8 SYSTEM TESTING

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and it's original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by giving input values to the system and by comparing with expected output. Top-down testing implementing here.

6.5 ACCEPTANCE TESTING

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

6.6 TEST CASES

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and with in their operational bounds.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

After implementing the website it will contain the advantages were incomparable to the present contemporary systems used by company. The most admirable feature founded was its simplicity in terms of application to the user but its highly beneficial outputs can't be ignored. The users will be highly benefited after using the system. It is hoped that this project will help the future developers to modify and implement the system. After modifying some techniques of the programs, it will give us the best performance as our requirements. The project will be very useful for the users.

7.1.1 FUTURE SCOPE

This application has wide range of scope in the upcoming era. Some of them are listed below:

- **Virtual Education:** During the unpredictable period as we faced such as lockdown, it is impossible to arrange the hard copies of the book so this type of website can reduce the barrier to get knowledge at any place in a cost effective, productive way.
- **E-library:** For students who are interested in learning online can use this website and keep all the books they want to learn from at one place (in favourites section) and can create their own personal E-library.
- **Imparting Education**: It Will complying with the new Educational Reforms and hence saving the paper that is used in making books.
- **Book stores**: Even individual book stores can have this system of book websites promoting their brand name as Digital Marketing and can gain number of customers.

APPENDIX 1

SOURCE CODES

HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Book Finder by Sanjay</title>
  <meta name="description" content="Find some books using Google's Book API">
  <meta name="keywords" content="Oleksandr, Malakhov, alexever17, books, API, coding, google">
  <meta name="author" content="Oleksandr Malakhov">
  <link rel="shortcut icon" type="image/x-icon" href="favicon.ico">
  <meta property="og:image" content="https://www.bookbusinessmag.com/thumb/?src=/wp-content/uploads/sites/4/2016/01/google-books-750x340.png&w=750&h=340">
  <link href="https://fonts.googleapis.com/css?family=Patrick+Hand" rel="stylesheet">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="assets/uikit-3/uikit.css" />
  <script src="assets/uikit-3/uikit.js"></script>
  <link rel="stylesheet" href="assets/master.css">
</head>
<body>
```

```
  <main>
    <h1>Book Finder </h1>
    <div class="uk-margin">
      <form class="uk-search uk-search-large" id="book-search">
        <a href="" class="uk-search-icon-flip" uk-search-icon id="search-button"></a>
        <input class="uk-search-input" type="search" placeholder=" Search..." id="search-
input">
      </form>
    </div>
  </main>
  <div id="result">
    <div>
      <div class="uk-card uk-card-default uk-card-hover uk-card-body">
        <h3 class="uk-card-title">Welcome!</h3>
        <p>Type in your search request and press the search icon. :)</p>
      </div>
    </div>
    <div>
      <div class="uk-card uk-card-default uk-card-hover uk-card-body">
        <h3 class="uk-card-title">Search</h3>
        <p>Hopefully, you will find something nice.:)</p>
      </div>
    </div>
```

```
  </div>
  <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
  <script src="assets/main.js"></script>
</body>
</html>
```

CSS

```
html {
  background-image: url("prism/prism/prism.png");
}

h1 {
  color: white;
}

h1, h2, h3, h4 {
  font-family: 'Patrick Hand', cursive !important;
  font-size: 3.3em;
}

main {
  text-align: center;
  padding: 3% 3% 0 3%;
}

#result {
  display: -webkit-box;
```

```
  display: -ms-flexbox;
  display: flex;
  -ms-flex-wrap: wrap;
      flex-wrap: wrap;
  -ms-flex-pack: distribute;
      justify-content: space-around;
  padding: 3%;
}
#result > * {
  -ms-flex-preferred-size: 49%;
      flex-basis: 49%;
  margin-top: 3%;
}

.uk-search {
  background-color: white;
}

.uk-search-icon-flip {
  color: #999999 !important;
}

@media only screen and (max-width: 700px) {
  #result > * {
    -ms-flex-preferred-size: 100%;
        flex-basis: 100%;
  }
```

```
}
```

SCSS

```
html {
   background-image: url("prism/prism/prism.png");
}
h1 {
  color:white;
}
h1, h2, h3, h4 {
  font-family: 'Patrick Hand', cursive !important;
  font-size: 3.3em;
}

main {
  text-align: center;
  padding: 3% 3% 0 3%;
}

#result {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
  padding: 3%;
  &>* {
    flex-basis: 49%;
    margin-top: 3%;
  }
```

```
}

.uk-search {
  background-color: white;
}

.uk-search-icon-flip {
  color: rgb(153, 153, 153) !important;
}

@media only screen and (max-width: 700px) {
  #result {
  &>* {
    flex-basis: 100%;
  }
}
}
```

JAVASCRIPT

```
const api_key = "AIzaSyDdkUJrGPPVWMRg-enDhs2Y_2xvkvQT6lI";
const search_button = document.getElementById("search-button");
const search_input = document.getElementById("search-input");
const displayArea = document.getElementById("result");
search_button.addEventListener("click", search);
document.addEventListener("keypress", function (e) {
  if (event.keyCode == 13) {
    search(e);
```

```
  }
})
function search(e) {
  e.preventDefault();
  if (search_input.value != "") {
    const raw = search_input.value.toLowerCase();
    const query = raw.replace(/\s/,"+");
    asyncCall(query);
  } else {
    alert("Please type in your search request")
  }
}
async function asyncCall(query) {
  try {
    let data = await
axios.get(`https://www.googleapis.com/books/v1/volumes?q=${query}s&key=${api_key}
`);
    display(data);
  } catch (err) {
    alert("Something went wrong. Please try again");
  }
}
function display(data) {
  dataArray = data.data.items
  let processedData = [];
  for (let i = 0; i < dataArray.length; i++) {
```

```
    if (dataArray[i].volumeInfo.authors == undefined) { dataArray[i].volumeInfo.authors = ["#Missing Entry#"]}
    if (dataArray[i].volumeInfo.publisher == undefined) { dataArray[i].volumeInfo.publisher = "#Missing Entry#" }
    const element = `
      <div class="uk-card uk-card-default uk-card-hover uk-grid-collapse uk-child-width-1-2@s" uk-grid>
        <div class="uk-card-media-left uk-cover-container">
          <img src="${dataArray[i].volumeInfo.imageLinks.thumbnail}" alt="The image to ${dataArray[i].volumeInfo.title} by ${dataArray[i].volumeInfo.authors[0]}" uk-cover>
          <canvas width="600" height="400"></canvas>
        </div>
        <div>
          <div class="uk-card-body">
            <h3 class="uk-card-title">${dataArray[i].volumeInfo.title}</h3>
            <p>Written by ${dataArray[i].volumeInfo.authors[0]}</p>
            <p>Published by ${dataArray[i].volumeInfo.publisher}</p>
            <a href="${dataArray[i].volumeInfo.previewLink}" class="uk-button uk-button-primary">Further Info</a>
          </div>
        </div>
      </div>
    `;
    processedData.push(element);
  }
  displayArea.innerHTML = "";
  displayArea.innerHTML = processedData.join("");
```

APPENDIX 2

SCREENSHOTS

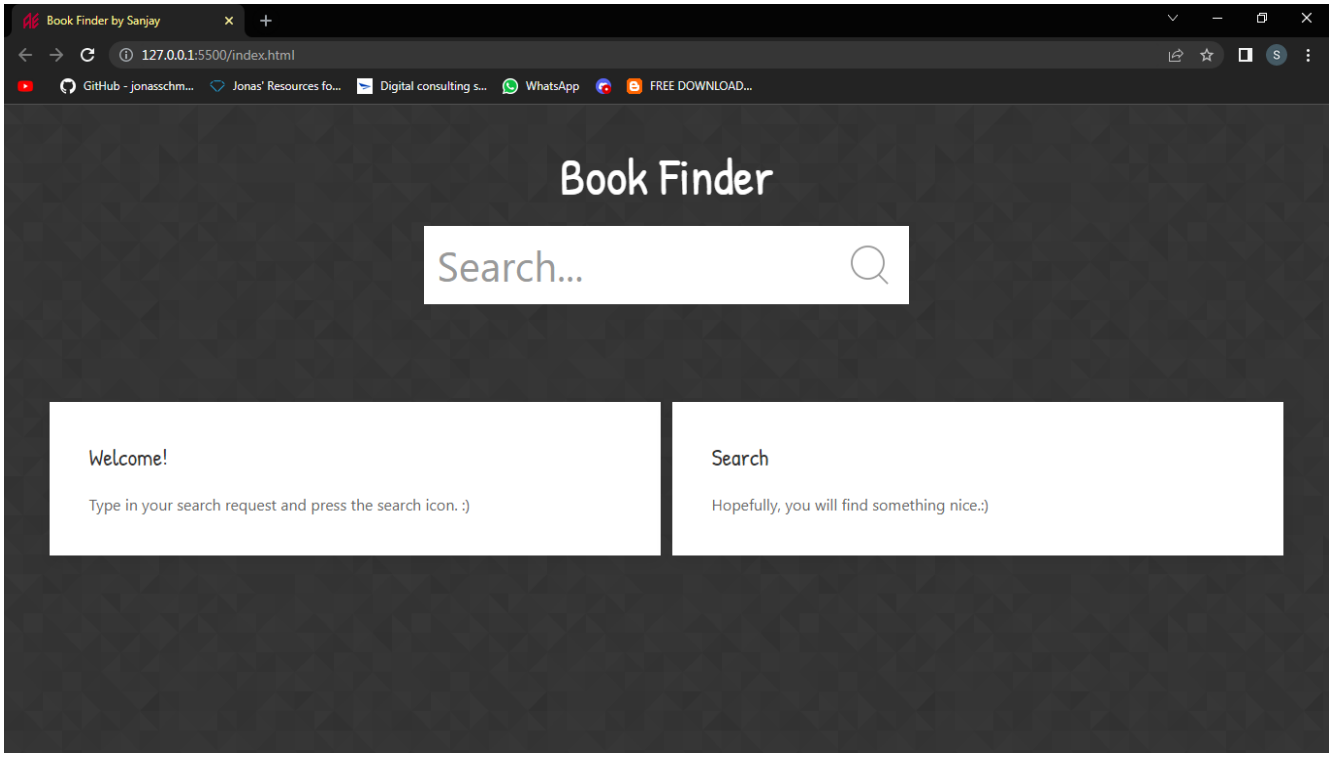

Figure A.2.1: HomePage

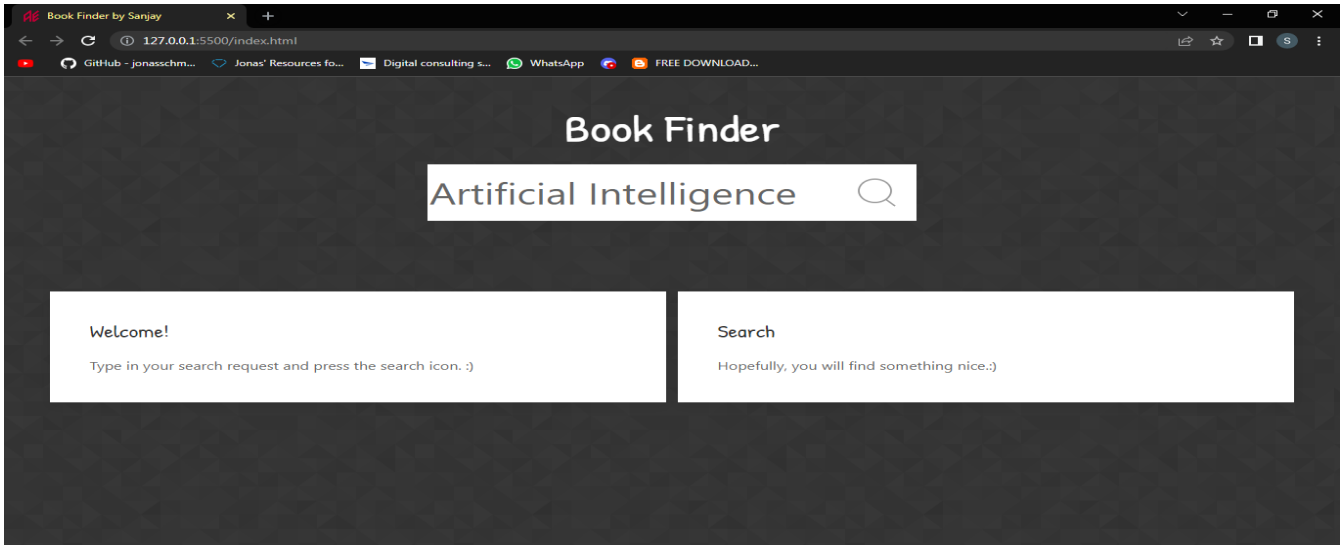

Figure A.2.2. Searching The Book

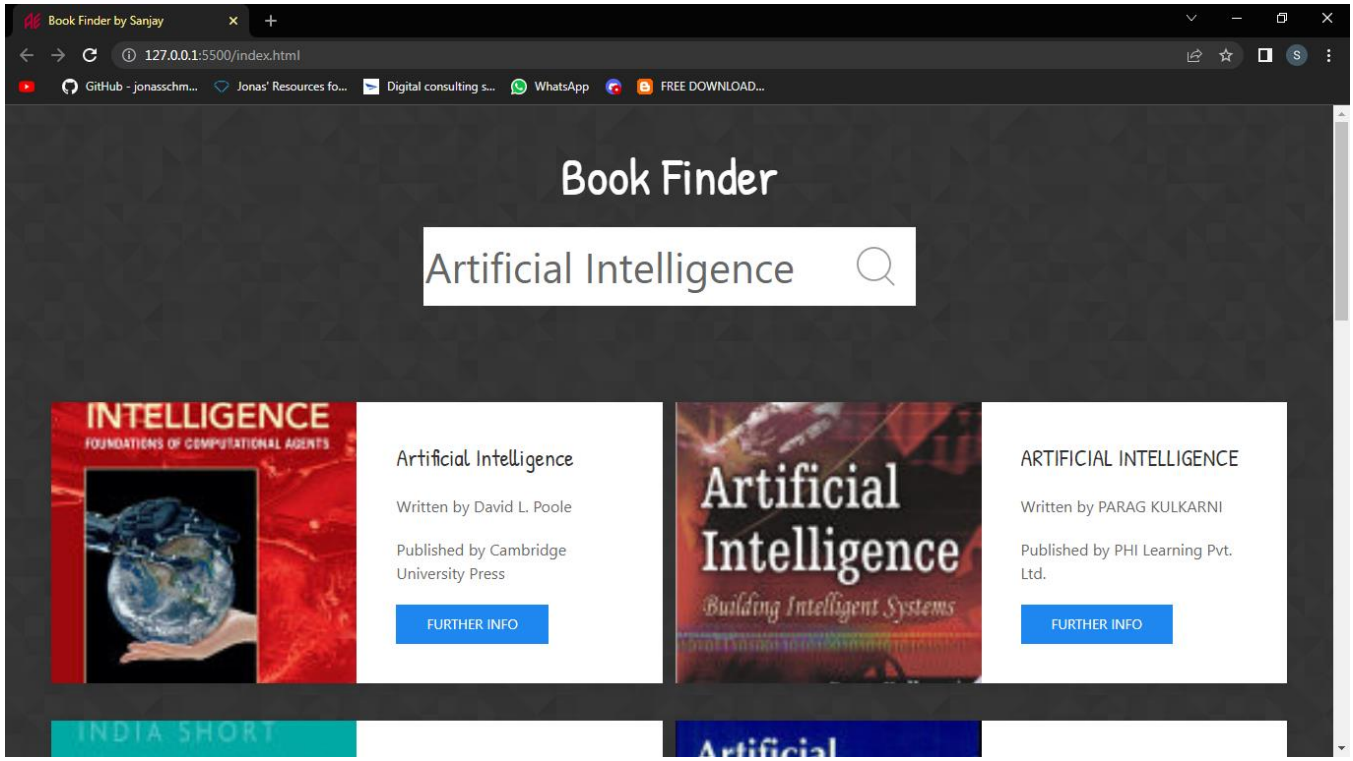

Figure A.2.3 : The Books Will Be ListOut

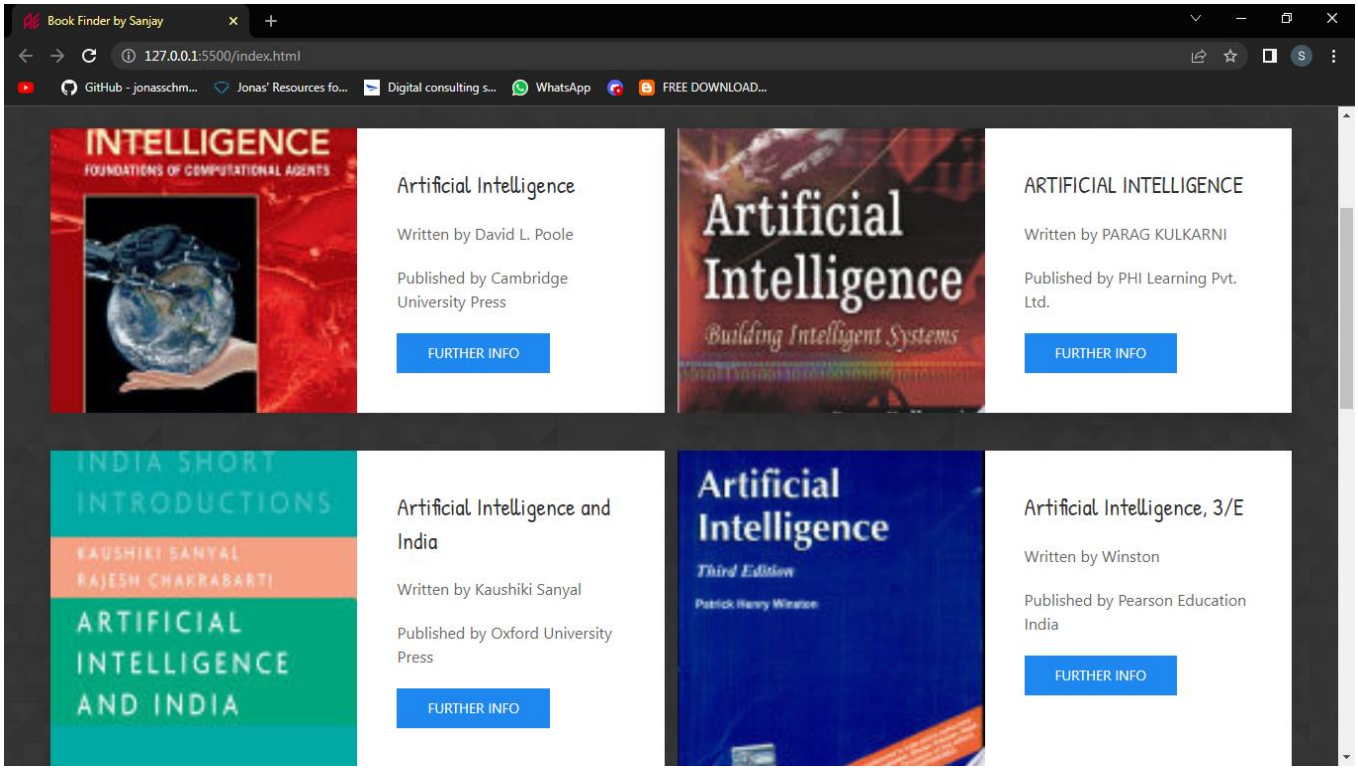

Figure A.2.4 : List Out Few Some Books

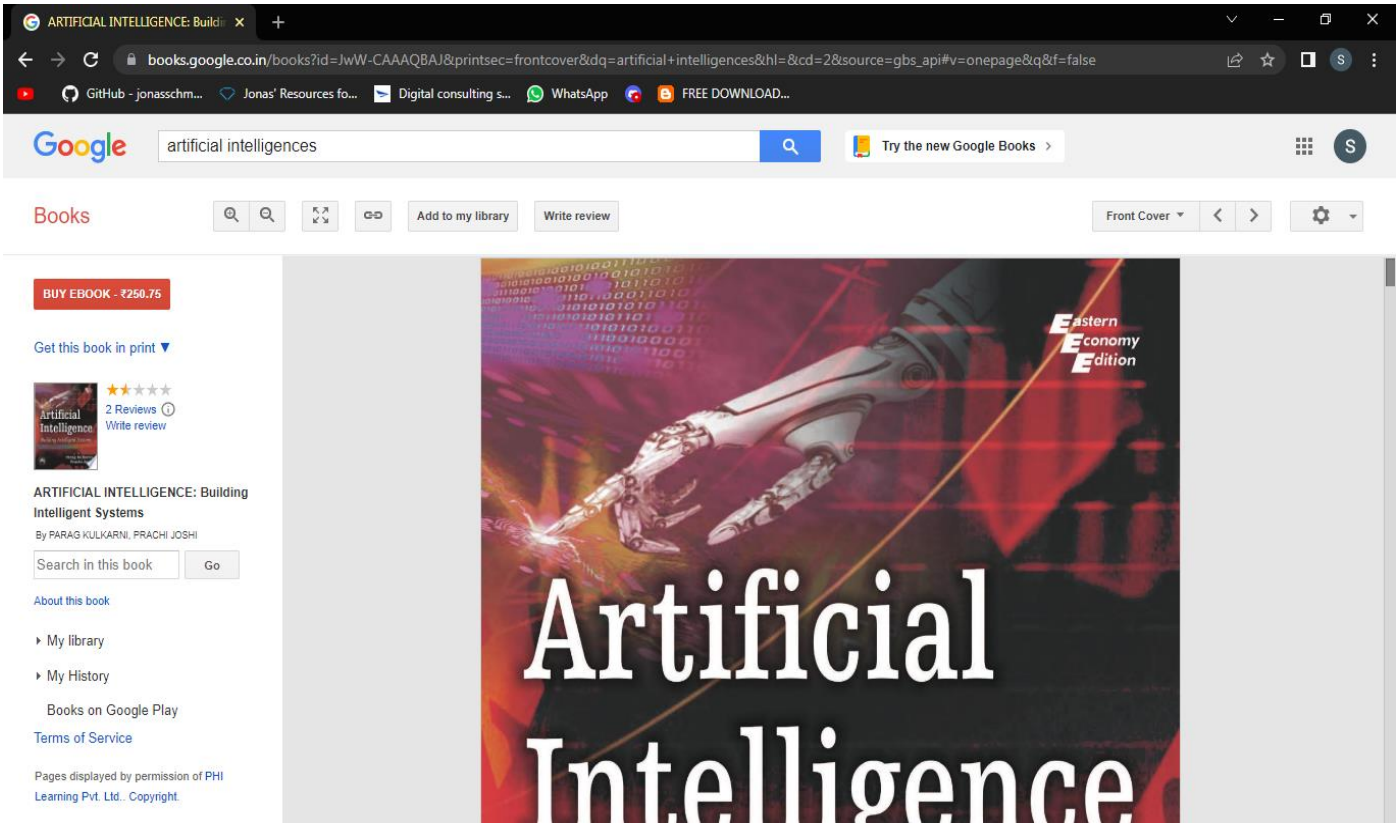

Figure A.2.5 : View The Books Review , Price,Buy The Book History and Library

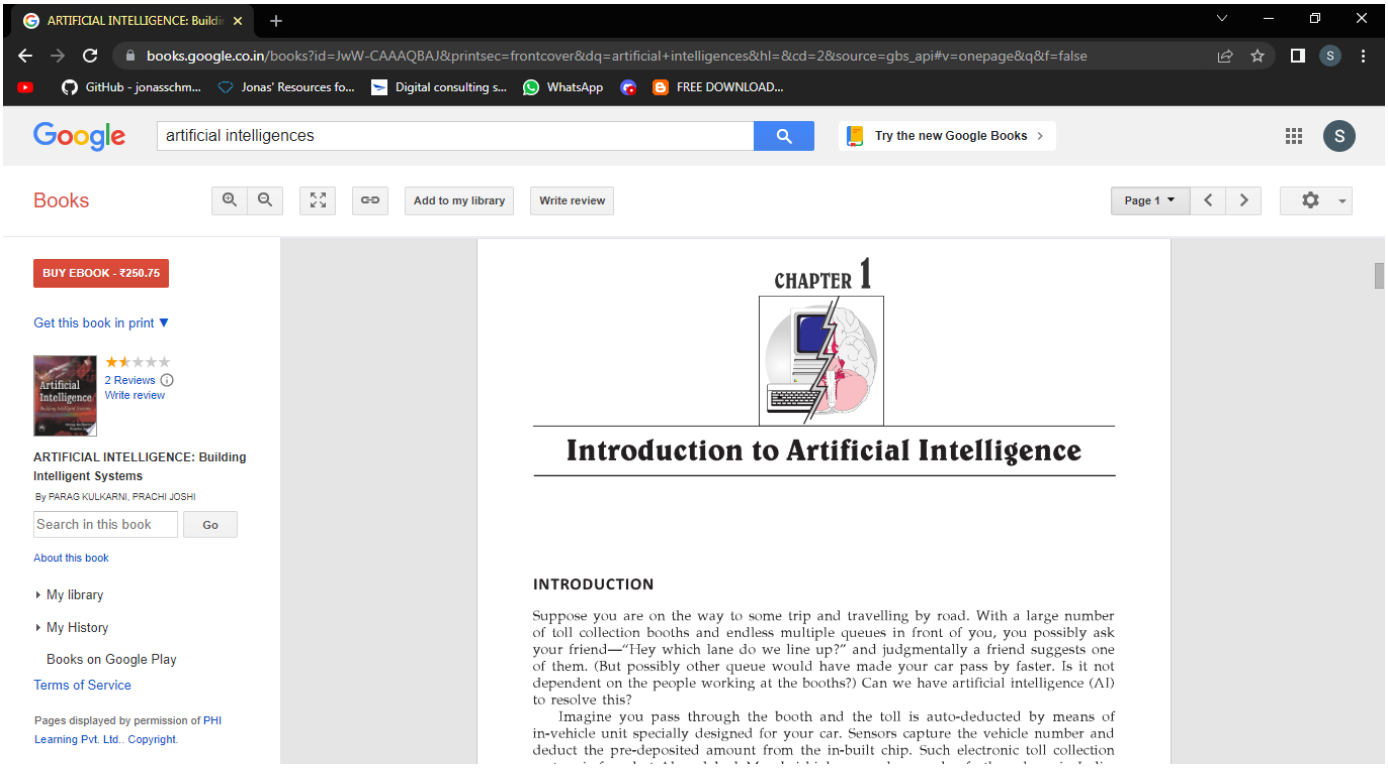

Figure A.2.6 : View The Book in Pdf Format

CHAPTER 8

REFERENCES

8.1 BOOKS & WEBSITE REFERED

[1] Kaneez Laila Bhatti1, Laraib Mughal1, Faheem Yar Khuhawar1,Sheeraz Ahmed Memon1,"Smart Attendance Management System Using FaceRecognition

https://eudl.eu/doi/10.4108/eai.13-7-2018.159713

[2] Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha, and Dwijen Rudra Pal,

"Study of Implementing Automated Attendance System Using Face Recognition Technique"

https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Nirmalya+Kar+& btnG=

[3] Bhautik Gondaliya, Shankar Sontakke, Kuldeep Suryawanshi, Tapsi Sonawane and Rupali Satpute," Attendance Management using Facial Recognition"

https://easychair.org/publications/preprint/Mfx2

[4] Neela Ashish Kumar, Swarnalatha P, Prudhvi Chowdary, Jai Krishna Naidu, Kailasa Sandeep Kumar,Undergraduate Students, School of Computer Science and Engineering, VIT," Smart Attendance Marking System using Facial Recognition"

https://rjstonline.com/HTML_Papers/Research%20Journal%20of%20Science%20and%20Technology__PID__2019-11-2-3.html

[5] V. Bhalla, T. Singla, A. Gahlot and V. Gupta, "Bluetooth based attendance management system", *International Journal of Innovations in Engineering and Technology (IJIET)*, vol. 3, no. 1, pp. 227-233, 2013.

[6] https://scholar.google.com/scholar?as_q=Bluetooth+based+attendance+management+system&as_occt=title&hl=en&as_sdt=0%2C31

[7] P. Viola and M. J. Jones, "Robust real-time face detection", *International journal of computer vision*, vol. 57, no. 2, pp. 137-154, 2004.

https://digital-library.theiet.org/content/journals/10.1049/iet-spr.2008.0172

[8] Naveed Khan Baloch, M. HaroonYousaf, Waqar Ahmad and M. Iran Baig, "Algorithm for Efficient Attendance Management: Face Recognition based Approach", *IJCSI*, vol. 9, no. 4, No 1, July 2012.

https://scholar.google.com/scholar?as_q=Algorithm+for+Efficient+Attendance+Management%3A+Face+Recognition+based+Approach&as_occt=title&hl=en&as_sdt=0%2C31

[9] Nirmalya Kar, Mrinal Kanti Debbarma, Ashim Saha and Dwijen Rudra Pal, "Study of Implementing Automated Attendance System using Face Recognition Technique", *IJCCE (International Journal of Computer and Communication Engineering)*, vol. 1, no. 2, July 2012.

https://scholar.google.com/scholar?as_q=Study+of+Implementing+Automated+Attendance+System+using+Face+Recognition+Technique&as_occt=title&hl=en&as_sdt=0%2C31