

E-Book Store

Semester- VII

Summer Internship Report

Submitted by

Name: Vyoma Suthar

Enrolment No: 210170107520

COMPUTER ENGINEERING DEPARTMENT

VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE

CHANDKHEDA



External Guide:

Mrs. Sweety Patel

**Learning and Development Executive
TATVASOFT, Ahmedabad**

Internal Guide:

Prof. Nakul R. Dave

**ASSISTANT PROFESSOR
VGEC, Chandkheda**

Gujarat Technological University

Academic Year 2023-24

Vishwakarma Government Engineering College, Chandkheda
Computer Department

CERTIFICATE



Date: 17/08/2023

This is to certify that the Summer Internship Report entitled E-Book Store Submitted by Enrolment No: **210170107520**, Name: **Vyoma Suthar** towards the fulfilment of Subject: Summer Internship (3170001) of Gujarat Technological University is the record of work carried out by him under our supervision and guidance in the Academic Year 2022-23.

Internal Guide:

Prof. Nakul R. Dave

Assistant Professor

VGEC Chandkheda

Head of Department

Prof M. T. Savaliya

Associate Professor

VGEC Chandkheda

Summer Internship Certificate

Date:18/08/2023

This is to certify that Vyoma Suthar undergone summer internship from 27th July 2023 to 10th August 2023. Details of the project is as under

Name of Project: Book Store Application

Technology: ReactJS

We wish him/her grand success for the future.

Authorized Signatory



TatvaSoft, Ahmadabad

ACKNOWLEDGEMENT

With great pleasure, I take this opportunity to express my deep sense of gratitude and indebtedness to my renowned and esteemed guide **Prof. Nakul R. Dave**, Assistant Professors, Department of Computer Engineering, Vishwakarma Government Engineering College, Chandkheda, for her consummate knowledge, invaluable guidance and encouragement which has enabled us to give present shape of this work.

I am heavily indebted to HOD **M.T. Savaliya**, Professor & Head, Department of Computer Engineering, Vishwakarma Government Engineering College, Chandkheda, for his everlasting willingness to extend his profound knowledge and experience in the preparation of this report. Any attempt to define this indebtedness would be incomplete.

Finally, I would like to thank our friends and family for their support and patience, and other faculty members of the department for their everlasting willingness to extend their support and help in the completion of this work. Especially to our parents who without their encouragement and financial support, this would not have been possible.

Yours Sincerely,
Vyoma Suthar
(210170107520)

ABSTRACT

As we know that Internet has become an inseparable part of our society; especially with the emergence of new technologies which are constantly changing the way business is being done. So, at Tatvasoft I was given a responsibility to develop an E-Book Store Web Application where users can buy and sell books and not just that, as there are plenty of E-Commerce Applications out there, my app is supposed to be as User Friendly as possible, also with nice User Interface with Optimized Features.

There are typically four major modules I had to work upon and they are Login Module, Registration Module, Products Module, Cart Module. Login and Registration Modules are used for Authentication; Products Module includes list of products, features to add product to cart and manipulate individual product information; Cart Module consists of feature to save cart and a list of products added to cart. Technologies that have been used to construct the whole Application are ReactJS, Formik, dotNET API and Postgres.

LIST OF FIGURES

Figure 1: Use Case Diagram.....	4
Figure 2: Class Diagram.....	5
Figure 3: System Architecture.....	6
Figure 4: Login Page.....	10
Figure 5: Home Page.....	10
Figure 6: Create an account Page.....	11
Figure 7: Book Listing Page.....	11
Figure 8: Search Book Page.....	12
Figure 9: Add to Cart Page.....	13
Figure 10: Cart Page.....	13

TABLE OF CONTENTS

Acknowledgement.....	I
Abstract.....	II
List of Figures.....	III
Table of Contents.....	IV
Chapter 1: Introduction.....	01
Chapter 2: System Analysis.....	02
2.1 Study of Current System.....	02
2.2 Problem and weakness of Current System.....	02
2.3 Requirement analysis of New System.....	02
2.4 Design: Analysis, Design Methodology and Implementation Strategy.....	03
Chapter 3: System Modelling.....	04
3.1 Use Case Diagram.....	04
3.2 Class Diagram.....	05
3.3 System Architecture Diagram.....	06
Chapter 4: Implementation.....	07
4.1 Day at TATVASOFT.....	07
4.1 Database Schema.....	08
4.2 Snapshots of Project.....	10
Conclusion and Future Scope.....	14
Plagiarism Report.....	15

CHAPTER 1: INTRODUCTION

‘E-Book Store’ website is an online book selling application which helps user to sell or buy books online. In today’s world, speed is everything. If the product that we are building for customers is not working properly such as if the system takes much more time to load than customers will not be satisfied. To solve that issue we have used the approach called ‘API first approach’.

In API first approach we divided the project into two parts. We separated both frontend and backend so that we don’t have to write same backend again. API is a URL endpoint. Whenever we want to change something in our database, we have to call that endpoint so that API endpoint will directly communicate with the database.

Here, ‘E-Book Store’ website uses the API first approach. For Frontend we have used the JavaScript based framework React and for backend API we have used ASP.net. To manage API endpoints, we have used the swagger UI and to store data in the database we have taken the help of Postgres.

In ‘E-Book Store’ website there are 5 important modules as stated below:

- User Login
- User Register
- Cart
- Book-listing
- Categories

In ‘E-Book Store’ website, there is a provision for creating a new account. Once account is created, user can login to the system. The home page contains lists of books. User also can search for any particular book by using search bar and finally, customer can add any book to the cart by clicking on the ‘Add to cart’ button.

Management of APIs is done by swagger UI which is pre-built component in ASP.net project. So, it is very easy to maintain API endpoints. We have built CRUD functionality for all the masters. Here, master means fully functional CRUD functionality.

Here, CRUD means Create, Read, Update, Delete. So, for every request we have separate endpoints.

In Frontend, form validation at run time is very important. When user send the data to backend, we have to check the user data format. If the data sent by the user in in wrong format, then the error message to the user is must.

For the purpose of form validation, we have used a very useful react package – Formik. But sometimes we have to write very much validation logic in Formic so another package call yup is very helpful. Yup has a pre-defined validation functions so it is very easy to write or learn or even understand.

CHAPTER 2: SYSTEM ANALYSIS

Before building a product for any problem, requirements of users must be clear in the mind of developers. The study of the current system is also a crucial thing as developers can know what's going on in the current system, technologies used, features, problems, and weaknesses of the current system.

2.1 Study of current system

In modern era, many companies are not using API first approach to build products. So, in that system both frontend and backend are together. When we have to test only frontend than it is very difficult to manage the project.

With the traditional system maintainability of the project will reduce and hard to achieve the efficiency.

2.2 Problems and Weakness of current System

- Cannot maintain it efficiently.
- We have to write separate backend for new platform.
- With the current system speed of website to load is low.
- Cost of project may increase due to this system.

2.3 Requirement analysis of New System

1. Functional Requirements

- Website shall have a home page that contains books list
- Website must have a user login / registration page.
- Website shall have a global search functionality.
- Website shall have an add to cart functionality.

2. Non-functional Requirements

- Performance
- Security
- Manageability

3. Hardware Requirements

- Windows 10 or later
- 8 Gb of RAM

2.4 Design: Analysis, Design, Methodology and Implementation Strategy

In this internship we are going to build this book selling website with API. So, for backend we are going to use ASP.net and for frontend we are going to use React. All the data will be stored in Postgres database and with the help of API, we are going to fetch the data from database and render the received data to the react frontend so that user can see the data.

With dotnet controllers we are going to make a backend CRUD APIs. And that API is directly connected to our database via models.

For frontend we are going to use a component-based structure because in react we don't have to write same component again and again. So, we have to reuse the components whenever we want.

In simple terms we are going to work on MVC idea. MVC means model, View and Controller.

Here, in our project React is responsible for UI components and it is working as **view**. Dotnet is working as **controllers** because all the request that is written is in dotnet and for **model**, there is Postgres database.

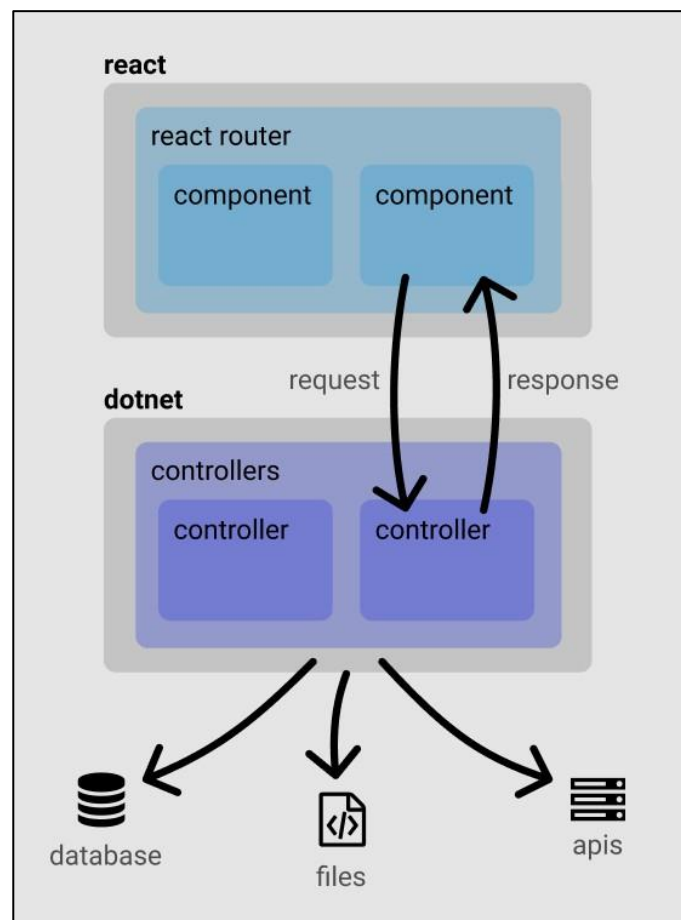


Fig. 2.4(a) Project Implementation Strategy

CHAPTER 3: SYSTEM MODELING

3.1 Use Case Diagram

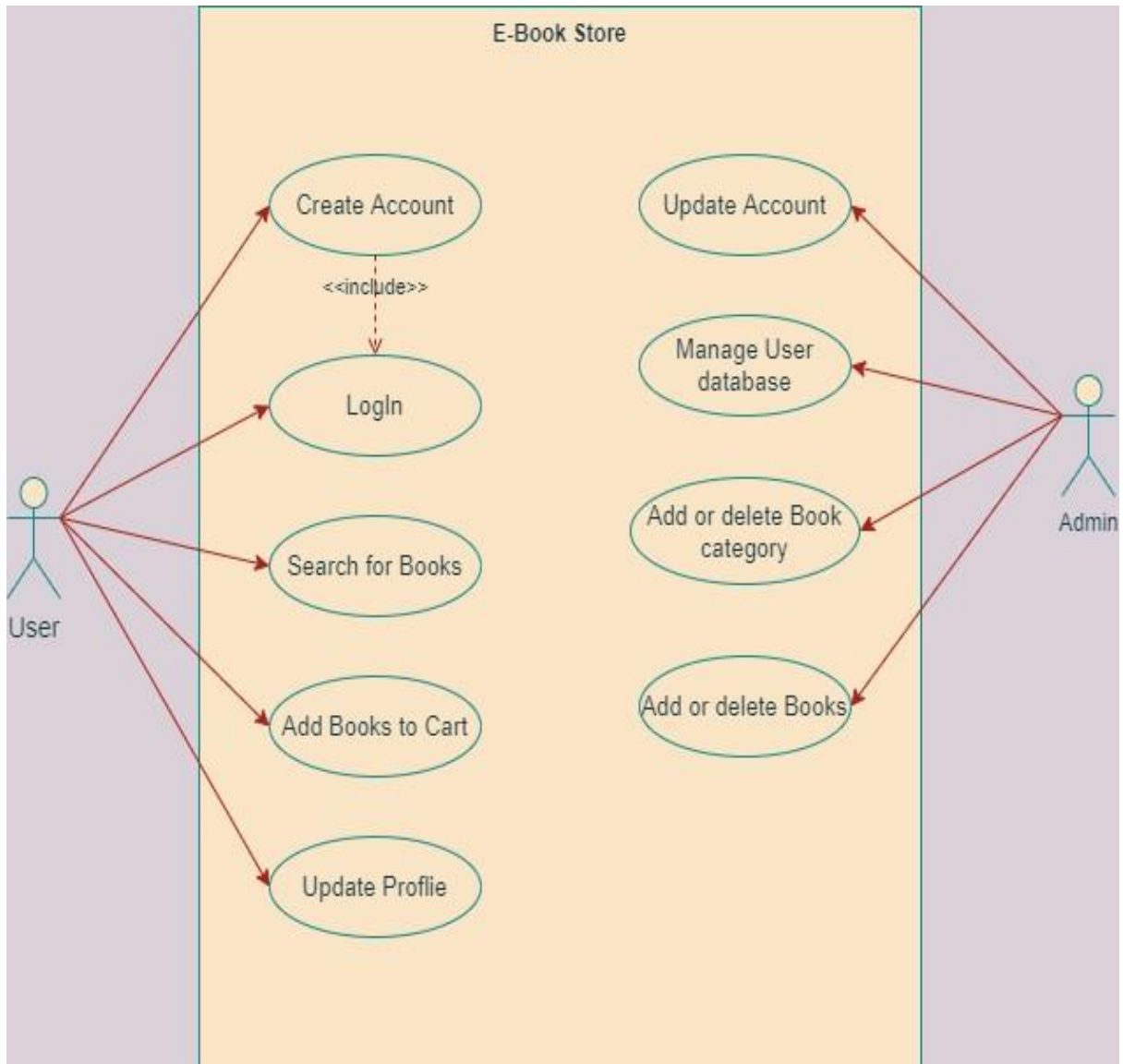


Figure 1: Use Case Diagram

3.2 Class Diagram

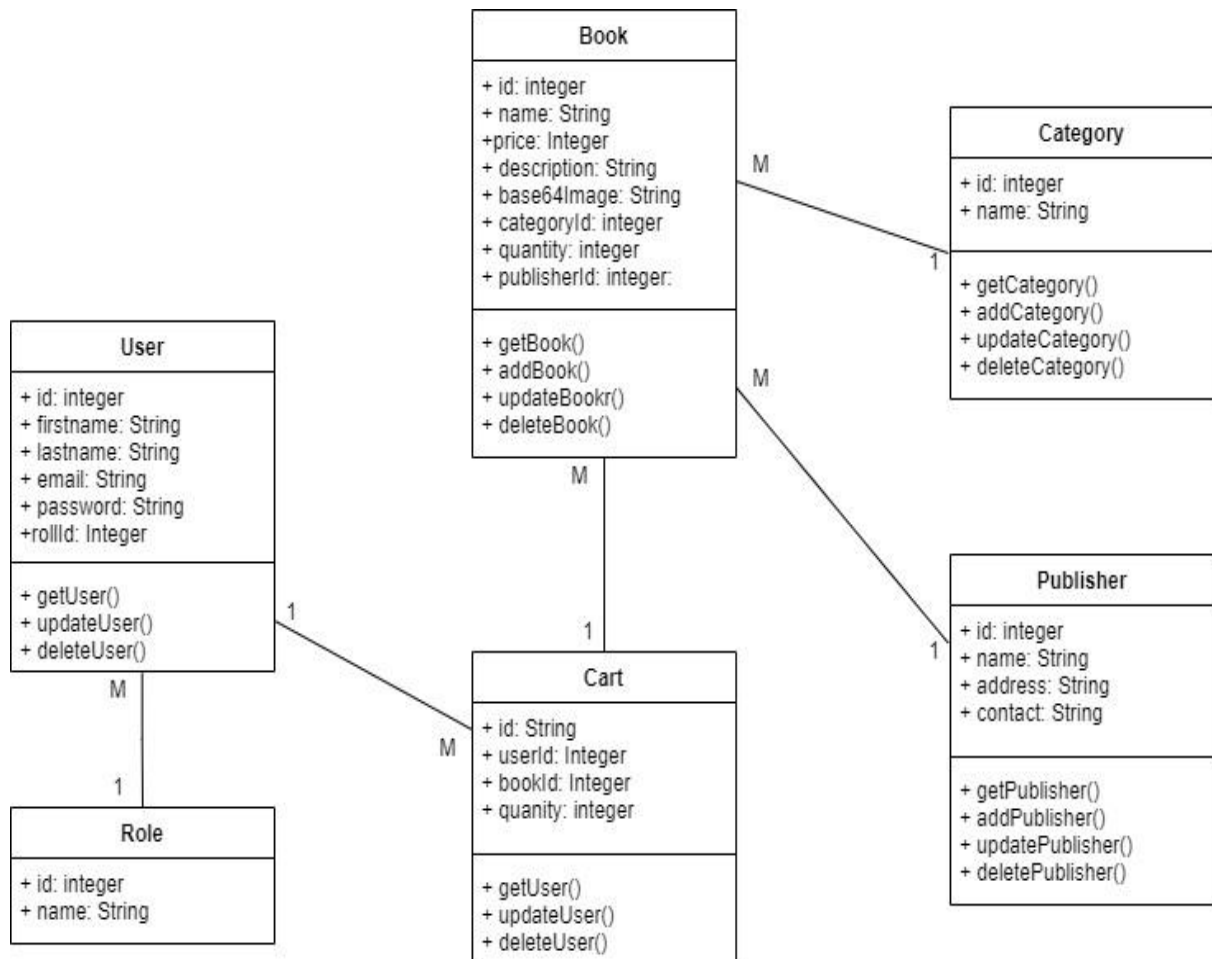


Figure 2: Class Diagram

3.3 System Architecture

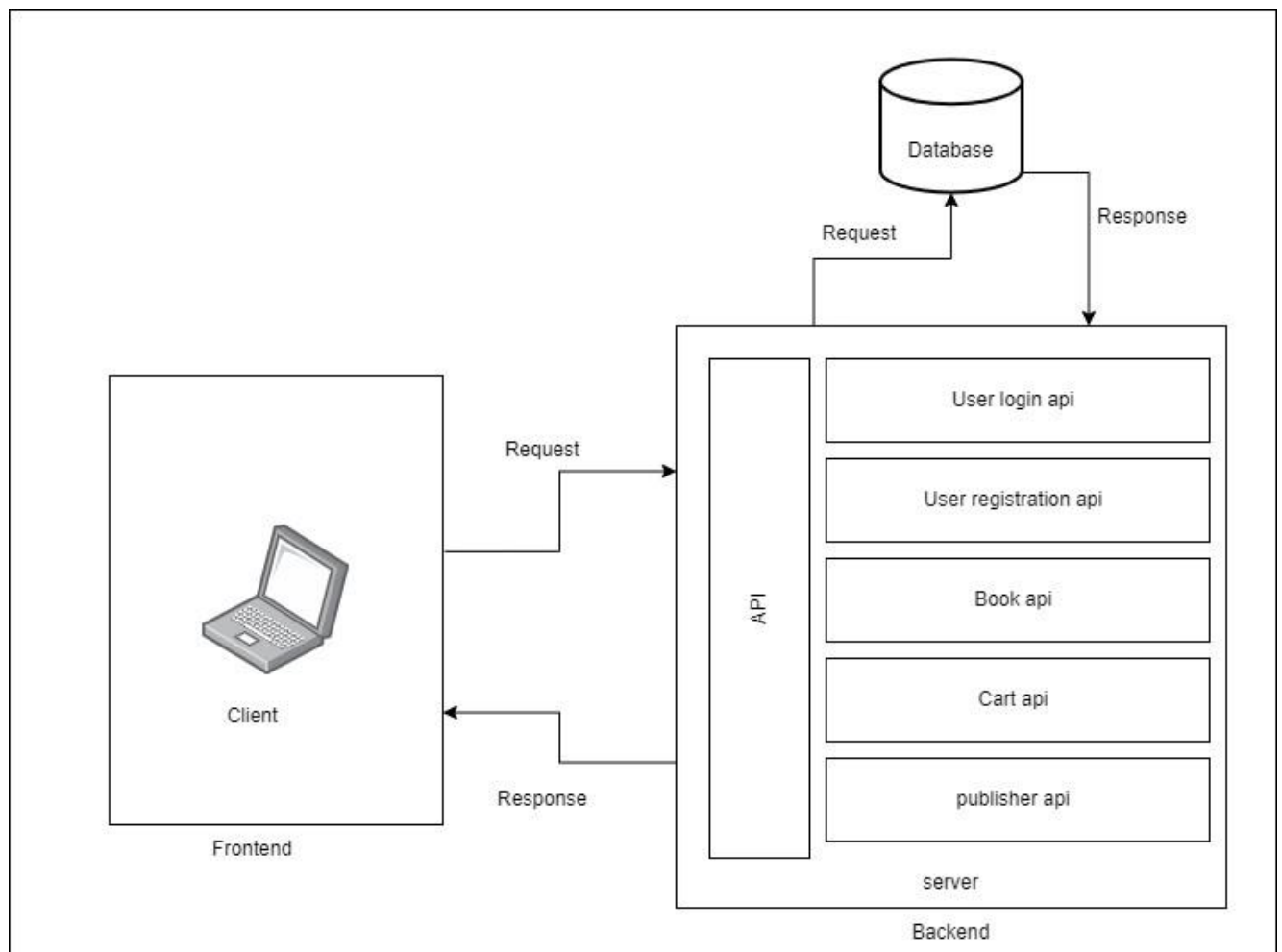


Figure 3: System Architecture

CHAPTER 4: IMPLEMENTATION

4.1 Day at TATVASOFT:

Tatvasoft offered online internship to us. So, we used to join on Microsoft Teams at 10 am and were given brief intro and list of tasks to perform on that particular day. We were encouraged to watch some videos of the topics that we are not aware of and perform the learning on our project. We used to have 3 different meetings in a particular day with the guide allocated to us who used to check and help us in implementation. We also used to have doubt solving session with the experts who used to solve our doubts if we had any. They used to analyse our work through screen sharing done by us. They also divided us into group of four so that we understand team building and help each other in performing tasks allocated.

We used to follow this routine daily. We had different task allocated which included learning as well as implementation. Every day they used to take daily update on the project and help us in implementing it if we had any issue.

E-Book Store Website:

Basically, my whole system is divided into 3 parts, view and controller.

1. Model

In my various data such as data of user, books, category, cart and publisher are stored in the Postgres database. Database stores important information about the all the tables like for User table there are different information such as id, firstname, lastname, email, password etc.

2. View

ReactJS works as View. View basically deal with how the data or page will display. React stores all the required information which is needed to be shown on screen in the component structure. Also, in React we can call all the required APIs which connects the frontend to backend.

3. Controller

CRUD APIs work as Controller. Using this APIs, it is possible to fetch required data from database and update those data if required. Dotnet has the support of swagger UI using which we can check the output from APIs.

Flow of My Project:

When user open my website, first, the home page will be rendered to the user screen. There are two options: login to the system and create new account. If the user is visiting the website for the first time, then they need to register to the system after that they will be able to access the site. Users who are already registered in the system need to login to access the system. If the entered credentials are valid, then, System will permit the user to login.

After login too the system, user will be able to see the main page of the site which will have books listed according to categories. User can find the appropriate for their interest by searching the book from the search bar from which they can search book by entering the name or the category. If a person wants to buy any book, then he can add that book to the cart by clicking on the ‘Add to Cart’ button. In the header, there is a cart button. If user clicks on the cart button, then, they will be redirected to the Cart page where they can find all the books they have added to the cart. Finally, user can place order by tapping on the Place Order button.

4.2 Data Dictionary

1. User

Field Name	Data Type	Field Size	Required
Id	Integer	-	Yes
Firstname	String	50	Yes
Lastname	String	50	Yes
Email	String	100	Yes
Password	String	100	Yes
Roleid	Integer	-	Yes

2. Role

Field Name	Data Type	Field Size	Required
Id	Integer	-	Yes
Name	String	50	Yes

3. Book

Field Name	Data Type	Field Size	Required
Id	Integer	-	Yes
Name	String	500	Yes
Price	Integer	10	Yes
Description	String	2000	No
Base64image	String	-	No
Categories	Integer	-	Yes
Publisherid	Integer	-	No
quantity	Integer	-	No

4. Cart

Field Name	Data Type	Field Size	Required
Id	Integer	-	Yes
Userid	Integer	-	Yes
Bookid	Integer	-	Yes
Quantity	Integer	-	Yes

5. Category

Field Name	Data Type	Field Size	Required
Id	Integer	-	Yes
Name	String	50	Yes

6. Publisher

Field Name	Data Type	Field Size	Required
Id	Integer	-	Yes
Name	String	50	Yes
Address	String	500	No
Contact	String	20	No

4.3 Snapshots of Project

1. Login Page

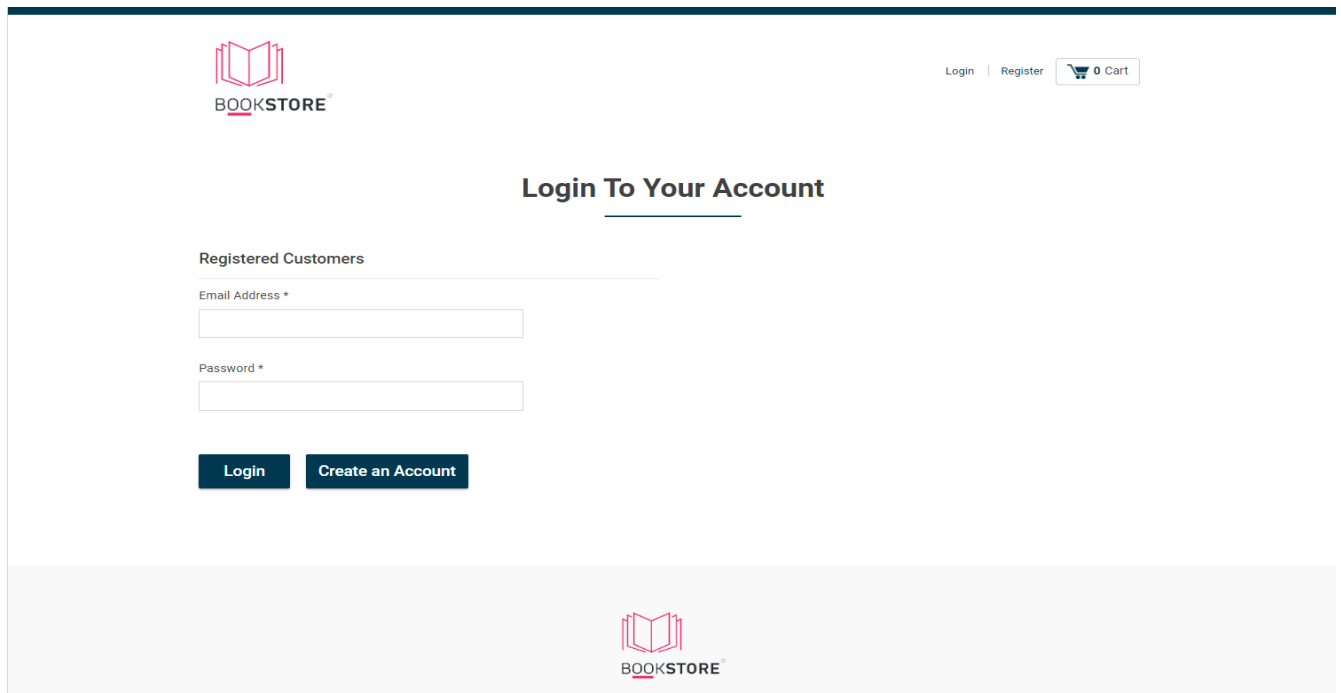


Figure 4: Login Page

When we click on the login page then then below screen appears.

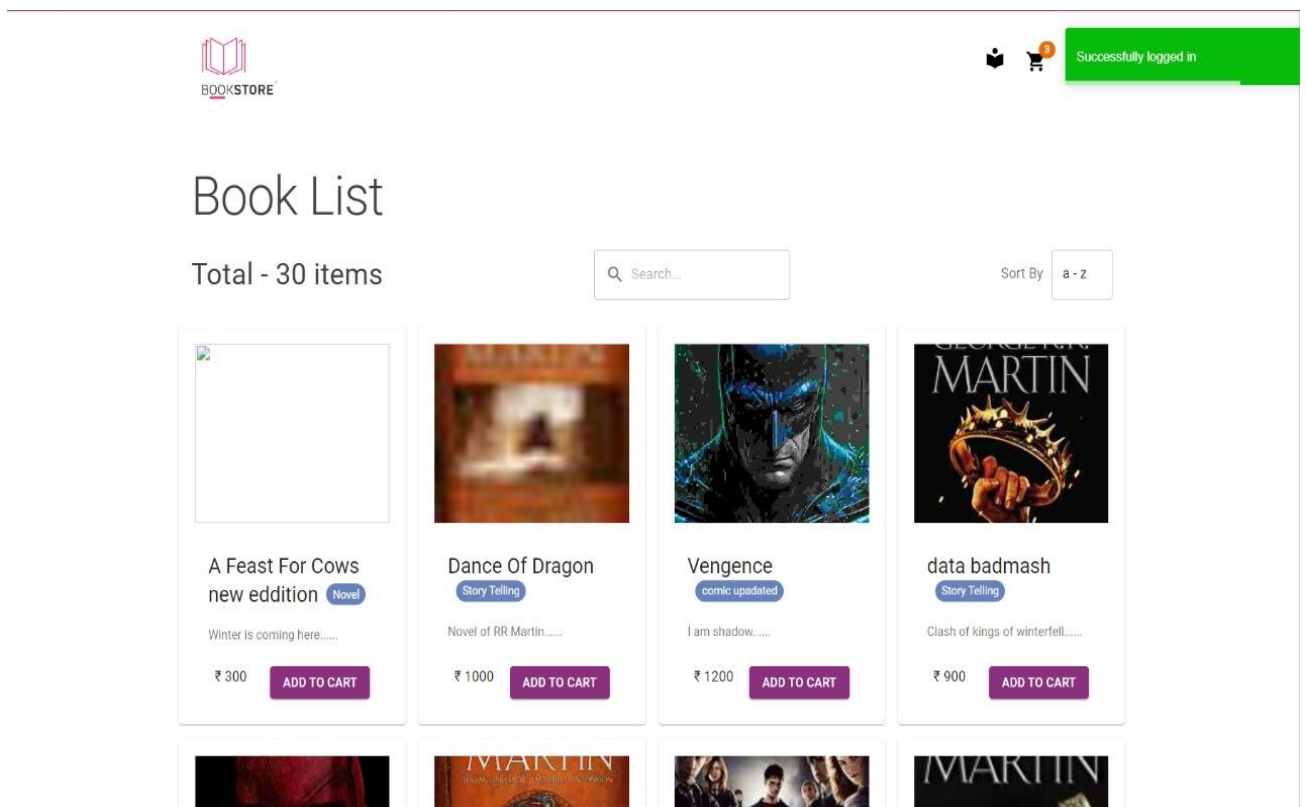




Figure 5: Home Page

2. Create Account Page




Login | Register  0 Cart

Create an Account

Personal Information

First Name *	Last Name *
<input type="text"/>	<input type="text"/>
Email Address *	Roles
<input type="text"/>	<input type="text"/>
Password *	Confirm Password *
<input type="text"/>	<input type="text"/>

Register



© 2023 bookstore. All rights reserved.

Figure 6: Create an Account Page

3. Book Listing Page

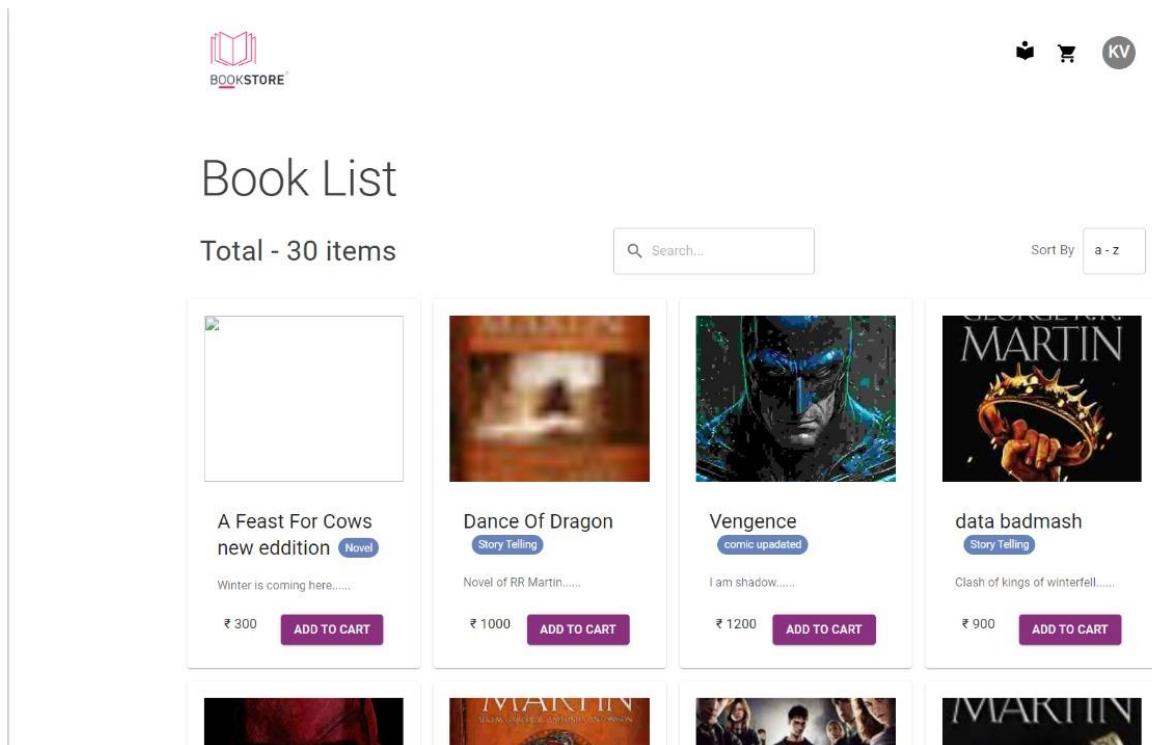


Figure 7:Book Listing Page

User can search for particular book by searching the name of the book in to search bar.

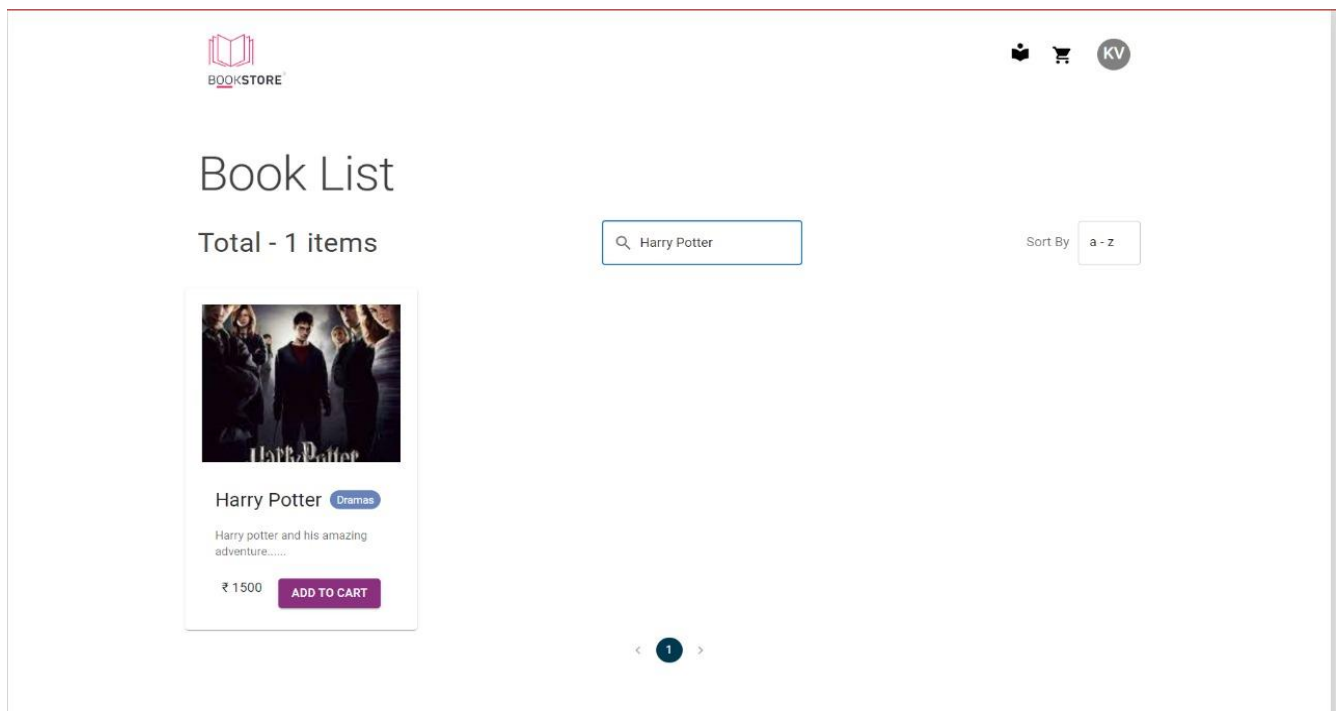


Figure 8: Search Book Page

4. Cart page

When User click on the add to cart button then one notification appears which informs User that that book is added to cart.

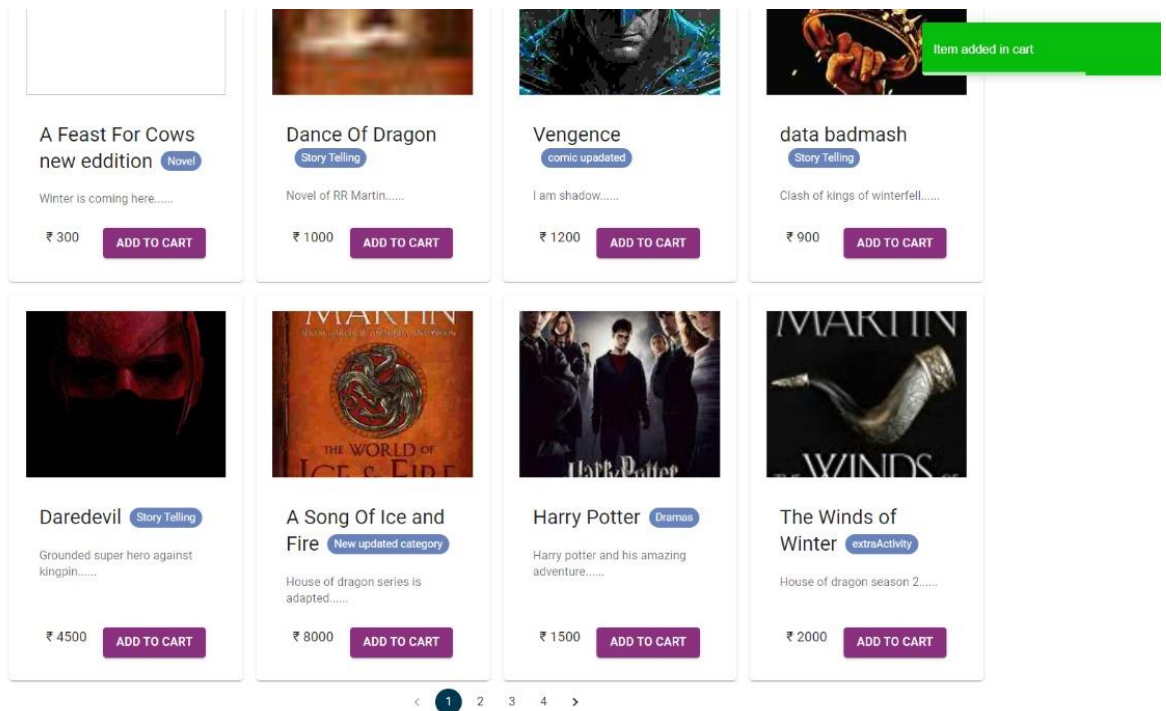


Figure 9: Add to Cart Page

All the item added to the cart will be visible in the cart page.

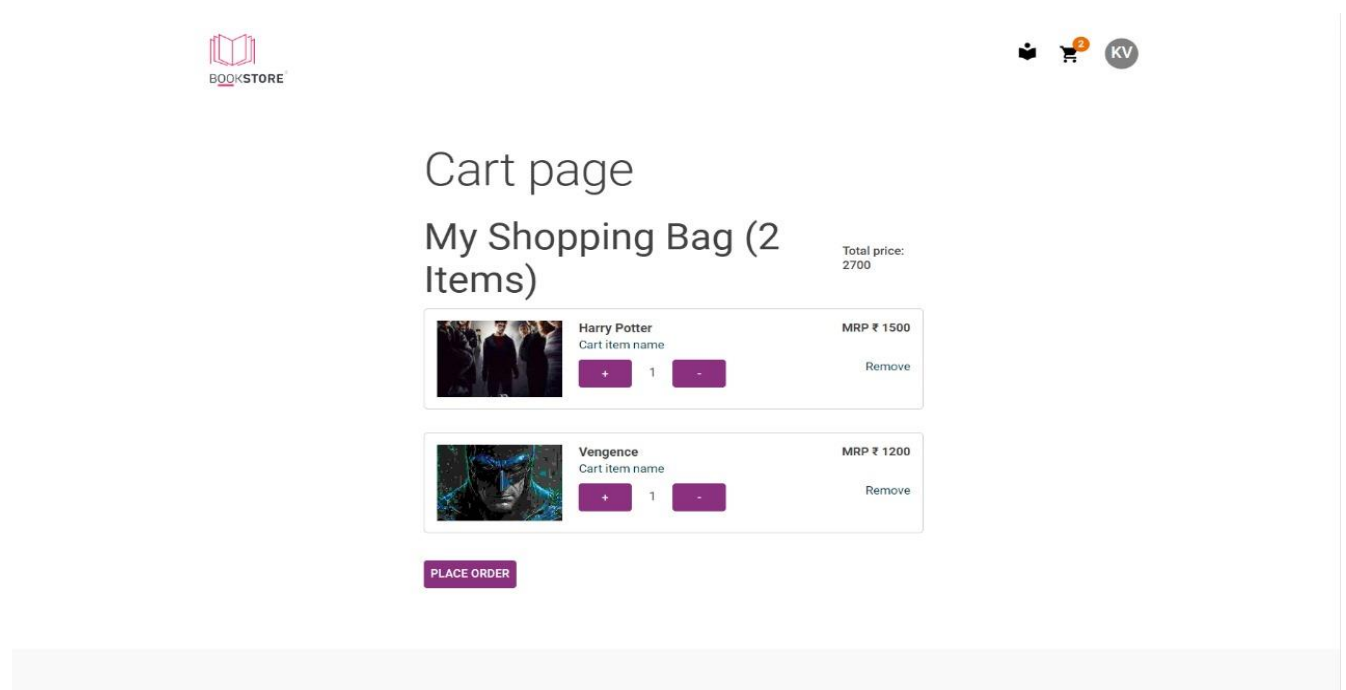


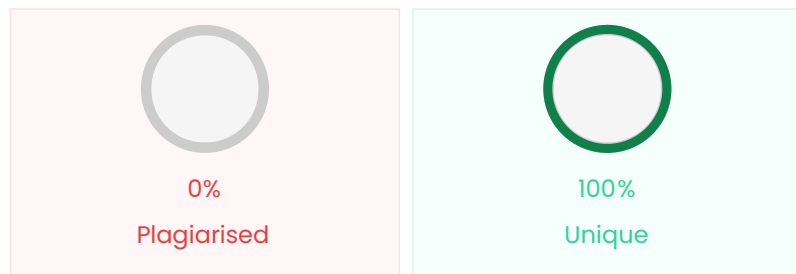
Figure 10: Cart Page

CONCLUSION AND FUTURE SERVICE

At last, I conclude that during these fifteen days of time we got to know about what kind of environment is there in the IT companies, how they coordinate with each other, how they improve themselves and how they maintain professionalism in their behaviour. We also learned about how the developers show professionalism in their work and how the work reflects the quality of developers.

As we know, the purpose of summer internship is to provide exposure to the students to the working environment of industries so that we can figure out how to present ourselves when we will be joining some industries. So finally, I would again thank TATVASOFT, for providing me such a wonderful opportunity of learning and working.

Plagiarism Scan Report



Words Statistics

Words	986
Characters	5635

Exclude URL

None

Content Checked For Plagiarism

CHAPTER 1: INTRODUCTION 'E-Book Store' website is an online book selling application which helps user to sell or buy books online. In today's world, speed is everything. If the product that we are building for customers is not working properly such as if the system takes much more time to load than customers will not be satisfied. To solve that issue we have used the approach called 'API first approach'. In API first approach we divided the project into two parts. We separated both frontend and backend so that we don't have to write same backend again. API is a URL endpoint. Whenever we want to change something in our database, we have to call that endpoint so that API endpoint will directly communicate with the database. Here, 'E-Book Store' website uses the API first approach. For Frontend we have used the JavaScript based framework React and for backend API we have used ASP.net. To manage API endpoints, we have used the swagger UI and to store data in the database we have taken the help of Postgres. In 'E-Book Store' website there are 5 important modules as stated below: • User Login • User Register • Cart • Book-listing • Categories In 'E-Book Store' website, there is a provision for creating a new account. Once account is created, user can login to the system. The home page contains lists of books. User also can search for any particular book by using search bar and finally, customer can add any book to the cart by clicking on the 'Add to cart' button. Management of APIs is done by swagger UI which is pre-built component in ASP.net project. So, it is very easy to maintain API endpoints. We have built CRUD functionality for all the masters. Here, master means fully functional CRUD functionality. Here, CRUD means Create, Read, Update, Delete. So, for every request we have separate endpoints. In Frontend, form validation at run time is very important. When user send the data to backend, we have to check the user data format. If the data sent by the user in in wrong format, then the error message to the user is must. For the purpose of form validation, we have used a very useful react package - Formik. But sometimes we have to write very much validation logic in Formic so another package call yup is very helpful. Yup has a pre-defined validation functions so it is very easy to write or learn

or even understand. CHAPTER 2: SYSTEM ANALYSIS Before building a product for any problem, requirements of users must be clear in the mind of developers. The study of the current system

Page 1 of 1

is also a crucial thing as developers can know what's going on in the current system, technologies used, features, problems, and weaknesses of the current system.

2.1 Study of current system

In modern era, many companies are not using API first approach to build products. So, in that system both frontend and backend are together. When we have to test only frontend than it is very difficult to manage the project. With the traditional system maintainability of the project will reduce and hard to achieve the efficiency.

2.2 Problems and Weakness of current System

- Cannot maintain it efficiently.
- We have to write separate backend for new platform.
- With the current system speed of website to load is low.
- Cost of project may increase due to this system.

2.3 Requirement analysis of New System

1. Functional Requirements

- website shall have a home page that contains books list
- website must have a user login / registration page
- website shall have a global search functionality
- website shall have a add to cart functionality

2. Non-functional Requirements

- performance
- security
- manageability

3. Hardware Requirements

- windows 10 or later
- 8 Gb of RAM

2.4 Design: Analysis, Design, Methodology and Implementation Strategy

In this internship we are going to build this book selling website with API. So, for backend we are going to use ASP.net and for frontend we are going to use React. All the data will be stored in Postgres database and with the help of API, we are going to fetch the data from database and render the received data to the react frontend so that user can see the data. With dotnet controllers we are going to make a backend CRUD APIs. And that API is directly connected to our database via models. For frontend we are going to use a component-based structure because in react we don't have to write same component again and again. So, we have to reuse the components whenever we want. In simple terms we are going to work on MVC idea. MVC means model, View and Controller. Here, in our project React is responsible for UI components and it is working as view. Dotnet is working as controllers because all the request that is written is in dotnet and for model, there is Postgres database.

Fig. 2.4(a) project Implementation Strategy

CHAPTER 3: SYSTEM MODELING

3.1 Use Case Diagram

3.2 Class Diagram

3.3 System ArchitectureCHAPTER 4: IMPLEMENTATION4.1 Day at Tatvasoft: Tatvasoft offered online internship to us. So, we used to join together on Microsoft Teams at 11 am and were given brief intro and list of tasks to perform in particular day. We were encouraged to watch some videos of the topics that we are not aware of and perform the learning on our project.



[Home](#)

[Blog](#)

[Testimonials](#)

[About Us](#)

[Privacy](#)

Copyright © 2022 Plagiarism Detector. All right reserved.