VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

A Mini Project Report on

ONLINE BOOK STORE MANAGEMENT

Submitted in partial fulfillment of the requirements as a part of the DBMS lab for the V semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

Submitted by

Subramanya M Rao - 1RN20IS163

Shodhan Shetty - 1RN20IS149

Under the Guidance of

Guide	Lab Incharge
Mr. Santhosh Kumar **Assistant Professor** **Dept. of ISE, RNSIT**	**Dr. R Rajkumar** **Associate Professor** **Dept. of ISE, RNSIT**

ESTD:2001

An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

2022 – 2023

RNS INSTITUTE OF TECHNOLOGY

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini project report entitled **ONLINE BOOK STORE MANAGEMENT SYSTEM** has been successfully completed by **SUBRAMANYA M RAO** bearing **USN 1RN20IS163** and **SHODHAN SHETTY** bearing **USN 1RN20IS149**, presently V semester student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree **Bachelor of Engineering in Information Science and Engineering** under **Visvesvaraya Technological University, Belagavi** during academic year 2022 – 2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

Mr. Santhosh Kumar	**Dr. R Rajkumar**	**Dr. Suresh L**
Assistant Professor	**Associate Professor**	Professor and HOD
Guide	Faculty Incharge	

External Viva

Name of the Examiners **Signature with date**

1. ____________________ ______________________________

2. ____________________ ______________________________

DECLARATION

We, **SUBRAMANYA M RAO [USN: 1RN20IS163**] and **SHODHAN SHETTY [USN: 1RN20IS149],** students of V Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Mini project entitled **ONLINE BOOK STORE MANAGEMENT SYSTEM** has been carried out by us and submitted in partial fulfilment of the requirements for the V Semester degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2022-2023.

Place: Bengaluru

SUBRAMANYA M RAO
[1RN20IS163]

SHODHAN SHETTY
[1RN20IS149]

ABSTRACT

The purpose of Online Book Store is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Online Book Store, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

ACKNOWLEDGEMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this project work.

We would like to express our thanks to our Principal **Dr. M K Venkatesha** for his support and inspired us towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. Suresh L**, Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our project work.

We would like to express our profound and cordial gratitude to our Faculty Incharge **Dr. R Rajkumar**, Associate Professor, Department of Information Science and Engineering for his valuable guidance, constructive comments and continuous encouragement throughout the mini project work.

We would like to express our profound and cordial gratitude to our Guide **Mr. Santhosh Kumar**, Assistant Professor, Department of Information Science and Engineering for his valuable guidance in preparing Project report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped us to carry out the project work.

And lastly, we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in carrying out this project work.

SUBRAMANYA M RAO (1RN20IS163)

SHODHAN SHETTY (1RN20IS149)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The "Online Book Store" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly.

Online Book Store, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Customer, Books, Order Status, Book Store, Stock. Every Online Book Store has different Book's needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

The admin can also add new book to the database. New books are brought into existence using this method. The admin has a specific password which can be changed once the position is occupied by a new individual.

Online Book Store also eliminates the delays in making the payments, where people use to stand for a long time in the queue to pay and get their book. It will also provide assurance that we have a proper record of the selling of the books so that we can limit our books stock according to that.

Chapter 2

LITERATURE SURVEY

2.1 Traditional File System

In the early days of computing, data management and storage were a very new concept for organizations. The traditional approach to data handling offered a lot of the convenience of the manual approach to business processes (e.g, handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom-built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

Separate information systems for each business function also led to conflicts of interest within the company. Departments felt a great deal of ownership for the data that they collected, processed, and managed which caused many issues among company-wide collaboration and data sharing.

2.1.1 Pros and Cons of the Traditional Approach

Pros

- **Simple**
  - Matched existing business processes and functions.
  - Companies were not as interested in funding complicated information systems
- **Initially low-cost**
  - Early computing was not viewed as beneficial for large funding.
  - Systems were designed to be cheap in order to save on cost.

Cons

- **Separated ownership**
  - Business functions had a high sense of data ownership.
  - Departments unwilling to share data for fear of minimizing their superiority.
- **Unmanaged redundancy**
  - Multiple instances of the same data appeared throughout various files, systems, and databases.
  - Information updated in one place was not replicated to the other locations.
  - Disk space was very expensive, and redundancy had a big impact on storage.
- **Data in consistency**
  - Redundant data stored in various locations was usually never stored the same way.
  - Formatting was not centrally managed.
- **Lack of data sharing**
  - Same data stored in multiple locations.
  - Caused unnecessary doubling of efforts for processing and managing data.
- **High costs in the long run**
  - Hiring data processors for each department was very expensive, and each position was typically working on the same thing just for a different area.
  - Doubling of work as well as excessive maintenance cost.

2.2 Downfall of Traditional Management System

Conceived in a relatively centralized era when software was deployed in static environments, legacy database architectures fail to support an increasingly mobile world where applications are accessed anytime, anywhere.

Today software users want consistent improvements in usability and expect SaaS vendors to deliver new features and functionalities needed to achieve their business objectives. However, legacy database technologies fall short. in serving the needs of todays distributedand cloud environments for the following reasons:

- Inadequate fail over capabilities
- Insufficient provisions during peak demands Latency issues
- Lack of high availability at all times Increasing operational costs
- Inability to meet the demands of global markets

For all of these reasons, traditional databases are unable to deliver results in a rapidly growing environment where the workload is geographically distributed across heterogeneous datacenters. Upgrading to a more distributed data model is costly and complicated and your DBAs can't just sit back and give up on this situation. Hence, due to these various reasons, the downfall of the traditional system was inevitable.

2.3 Introduction to the Database Management System

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of Data. Meaningful data combine to form Information. Hence, information is interpreted data- data provided with semantics.MS ACCESS is one of the most common examples of database management software. Database systems are meant to handle large collection of information Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crash or attempts at unauthorized access.

2.4 Indicative areas for the use of a DBMS

- Airlines: reservations, schedules etc.
- Telecom : calls made, customer details, network usage etc. Universities : registration, results, grades, etc.
- Sales: products, purchases, customers etc. Banking: all transactions

2.5 Advantages of a DBMS-

A Database Management System has many advantages over the traditional file system used in the earlier days, such as:

- **Data independence**: Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
- **Efficient data access**: DBMS utilize a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
- **Data integrity and Security** : If data is accessed through the DBMS , the DBMS can enforce integrity constraint on the data
- **Data administration**: When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation toreduce redundancy and make the data to retrieve efficiently

2.6 Components of a DBMS

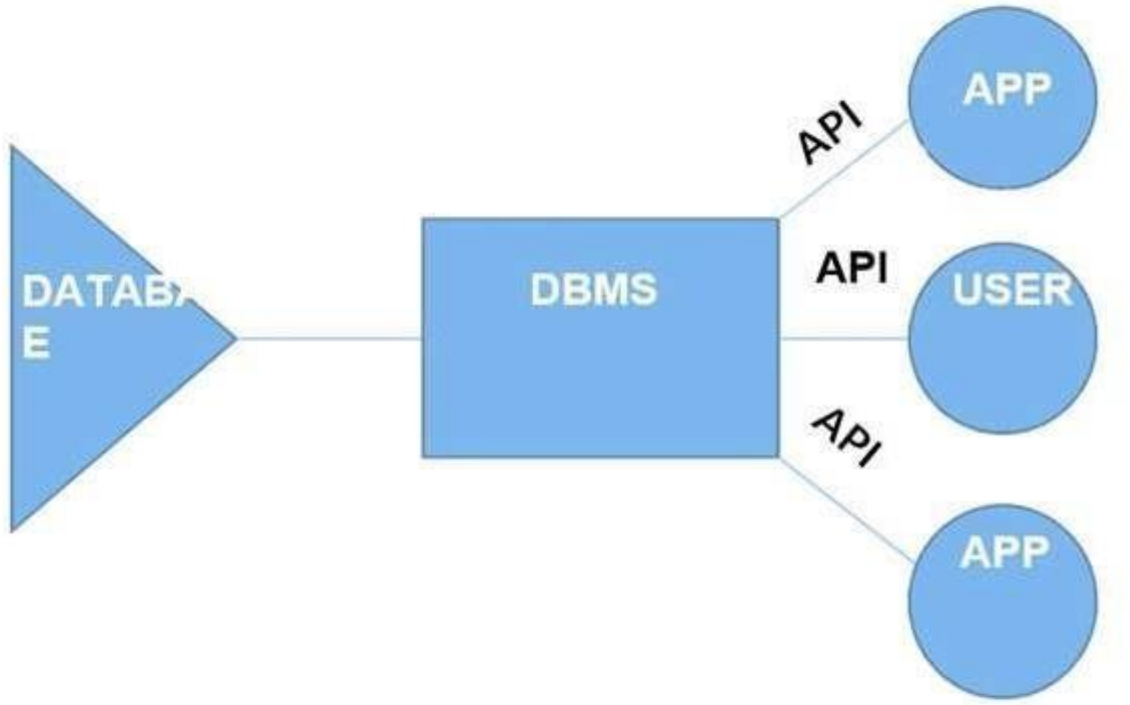


Figure 2.1 Components of DBMS

- **Users**: Users may be of any kind, such as data base administrators, system developers or database users.
- **Database application**: Database application may be Departmental ,Personal, Organizational and /or Internal
- **DBMS**: Software that allows users to create and manipulate database access.
- **Database**: Collection of logical data as a single unit.

Chapter 3

SYSTEM REQUIREMENTS

The main purpose of this SRS document is to illustrate the requirements of the project Student information System and is intended to help any organization to maintain and manage its student's personal data.

3.1 Hardware Requirements

Processor	: Intel Core i5
RAM	: 4GB
Hard Disk	:1TB

3.2 Software Requirements

Technologies Used:

Front End	: HTML5, CSS,BOOTSTRAP
Connection/Controller	: PYTHON-DJANGO
Back-End Database	: SQLite
Text Editor	: VS CODE
Server	: DJANGO server
Operating System	: Windows10/Linux/MAC
Database Support	: SQLite3
Back-End	: DJANGO

Chapter 4

SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

4.1 Entity Relation Diagram

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database as shown in figure.

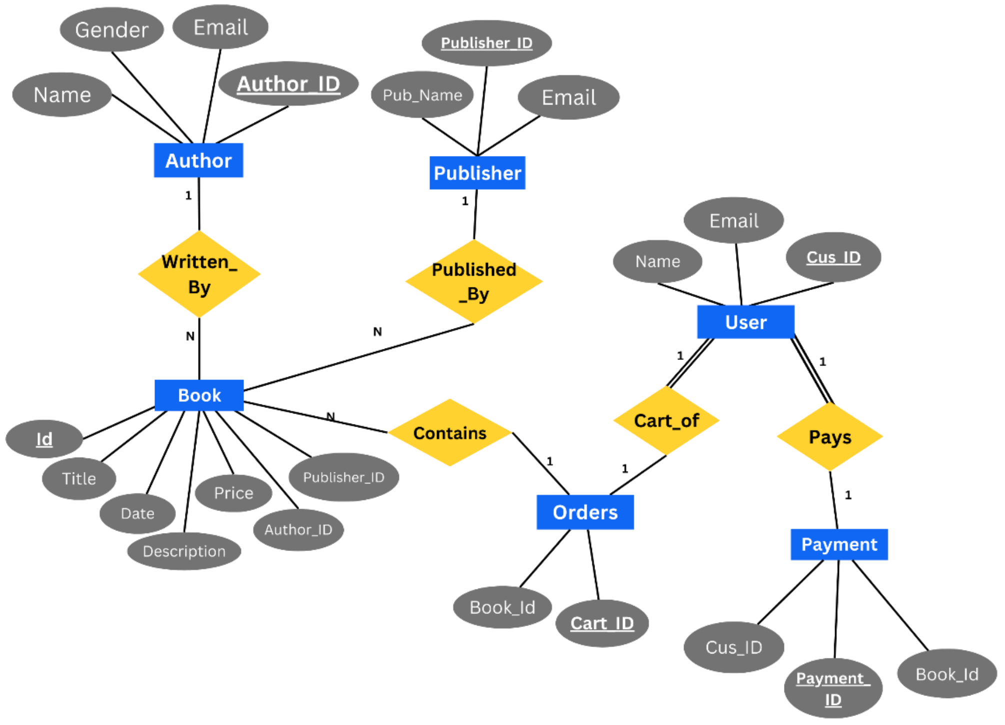


Figure 4.1 Entity Relational Diagram

4.2 Schema Diagram

A schema contains schema objects, which could be tables, columns, data types, store procedures, relationships, primary keys, foreign keys. A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

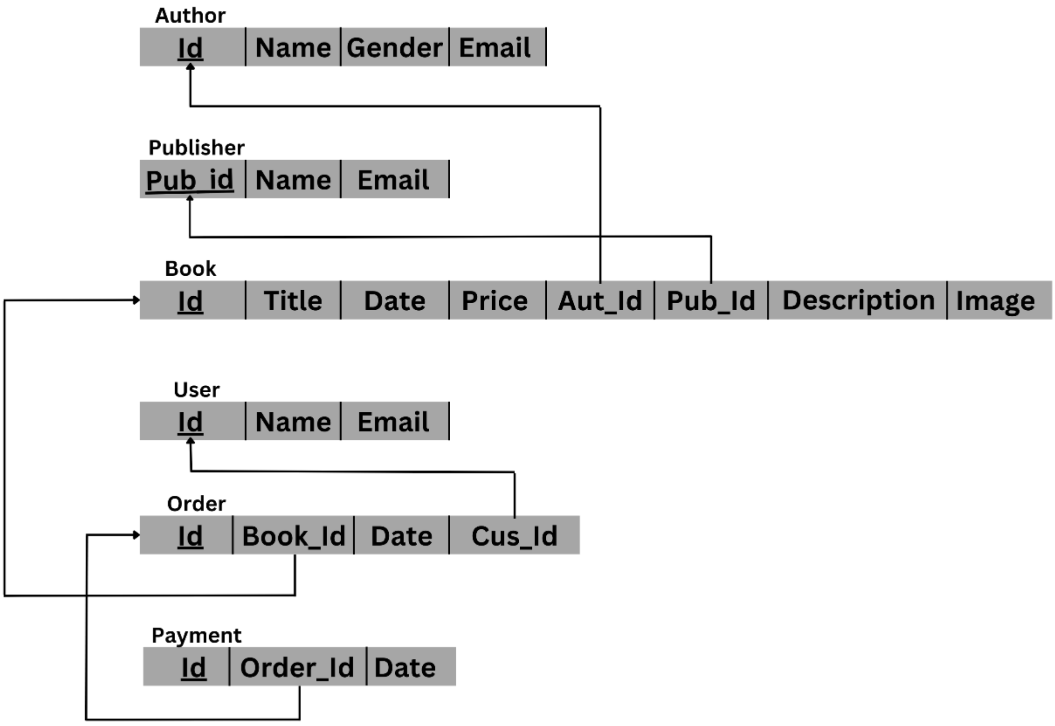


Fig 4.2 Schema diagram of Book Store Management System

Chapter 5

IMPLEMENTATION

5.1 HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard. It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc.HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2HTML.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as<main>, <section>, <article>,<header>,
<footer>, <aside>, <nav> and <figure>, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite> and<menu> have been changed, redefined or standardized.

The APIs and Document Object Model (DOM) are now fundamental parts of the HTML5 specification and HTML5 also better defines the processing for, any invalid documents.

5.2 DJANGO

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source. Django includes dozens of extras to handle common web development tasks. Django takes care of user authentication, content administration, site maps, and many more tasks- right out of the box.

Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords. Django for web development can be used in the following cases:

- For developing a Web Application or API Backend.
- For Rapid Development of some web application.
- Deploying the application Fast and Scaling it according to your needs
- A Perfect ORM for working with databases instead of database queries
- To develop a secure single-page application for either retrieving data or posting data.

5.3 SQLite (Structured Query Language)

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. SQLite database files are a recommended storage format by the US Library of Congress. Think of SQLite not as a replacement for Oracle but as a replacement for fopen().

SQLite is a compact library. With all features enabled, the library size can be less than 750KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger). There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be faster than direct filesystem I/O.

5.4 Code Snippets

5.4.1 Connectivity to Database

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),

    }

}
```

5.4.2 Insert a Record in Student Admission

```
publisher = Publisher(name='Publisher 1', email='publisher@example.com')

publisher.save()

author = Author(name='Author 1', gender='M', email='author@example.com')

author.save()

book = Book(title='Book 1', author=author, publisher=publisher, price=10.99)

book.save()


 class BookTest(TestCase):

     def setUp(self): # filling user data (credentials)

         self.user = get_user_model().objects.create_user(

         username = 'shodhan',

         email = 'shodhanshetty144@gmail.com',

         password = 'secret')

         self.book = Book.objects.create(   # filling Book model fields

         title = 'django for beginners',
            author = 'WS Vincent',
            description = 'anything',
            price = '30',
            image_url = 'https://forexample.jpg',
            follow_author = 'https://twitter.com/wsv3000?lang=en',
            book_available = 'True')

        def test_string_representation(self):
          book = Book(title='new book')
          self.assertEqual(str(book), book.title)

        def test_book_model_fields_content(self):

          self.assertEqual(f'{self.book.title}', 'django for beginners')

          self.assertEqual(f'{self.book.author}', 'WS Vincent')

          self.assertEqual(f'{self.book.description}', 'anything')

          self.assertEqual(f'{self.book.price}', '30')

          self.assertEqual(f'{self.book.image_url}', 'https://forexample.jpg')

          self.assertEqual(f'{self.book.follow_author}', 'https://twitter.com/wsv3000?lang=en')

          self.assertEqual(f'{self.book.book_available}', 'True')
```

5.4.3 View the contents of Students

```
class BooksListView(ListView):

  model = Book

  template_name = 'list.html'

class BooksDetailView(DetailView):

  model = Book

  template_name = 'detail.html'
```

```
class SearchResultsListView(ListView):
    model = Book
    template_name = 'search_results.html'

    def get_queryset(self):

        query = self.request.GET.get('q')

        return Book.objects.filter(

        Q(title__icontains=query) |

        Q(author__name__icontains=query) |

         Q(publisher__name__icontains=query)

         ).order_by('title')


class BookCheckoutView(LoginRequiredMixin, DetailView):

    model = Book

    template_name = 'checkout.html'

    login_url     = 'login'


def paymentComplete(request):

         body = json.loads(request.body)

         print('BODY:', body)

         product = Book.objects.get(id=body['productId'])

         Order.objects.create(product=product)

         return JsonResponse('Payment completed!', safe=False)
```

5.4.4 Admin Login

```
 AUTH_PASSWORD_VALIDATORS  = [
      {
      'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
      },
      {
      'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
      },
      {
       'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
      },
      {
      'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
      },
      ]

class PublisherAdmin(admin.ModelAdmin):
                list_display = ('name', 'email')



class AuthorAdmin(admin.ModelAdmin):
                list_display = ('name','gender', 'email')



class BookAdmin(admin.ModelAdmin):
                 list_display = ('title', 'author', 'publisher', 'price', 'book_available')



class OrderAdmin(admin.ModelAdmin):
                 list_display = ('product', 'created')



admin.site.register(Publisher, PublisherAdmin)
admin.site.register(Author, AuthorAdmin)
admin.site.register(Book, BookAdmin)
admin.site.register(Order, OrderAdmin)
```

5.5 Stored Procedure

SQLite does not provide the stored procedure concept; basically, stored procedure means we can prepare a single code, and that code we can reuse again and again as per user requirement.

A procedure in SQL (often referred to as stored procedure), is a reusable unit that encapsulates the specific business logic of the application. A SQL procedure is a group of SQL statements and logic, compiled and stored together to perform a specific task.

However we have created a reusable code which updates the database when data is added through admin page.

Code snippet

```
Procedure.CreateModel(
    name="Book",
    fields = [
      ("id",
         models.AutoField(
          auto_created=True,
          primary_key=True,
          serialize=False,
          verbose_name="ID",),
      ),
      ("title", models.CharField(max_length=200)),
      ("description", models.CharField(default=None, max_length=500)),
      ("price", models.FloatField(blank=True, null=True)),
      ("image_url", models.URLField(default=False, max_length=2083)),
      ("follow_author", models.URLField(blank=True, max_length=2083)),
      ("book_available", models.BooleanField(default=False)),
      ("author",
        models.ForeignKey(
          on_delete=django.db.models.deletion.CASCADE, to="books.author"),
      ),
    ],
  )
```

5.6 Trigger

Trigger is a statement that a system executes automatically when there is any modification to the database. In a trigger, we first specify when the trigger is to be executed and then the action to be performed when the trigger executes. Triggers are used to specify certain integrity constraint and referential constraints that cannot be specified using the constraint mechanism of SQL.

In our application whenever a user successfully checks out of the cart through PayPal payment gateway, a trigger will be called that updates the Orders table in our database

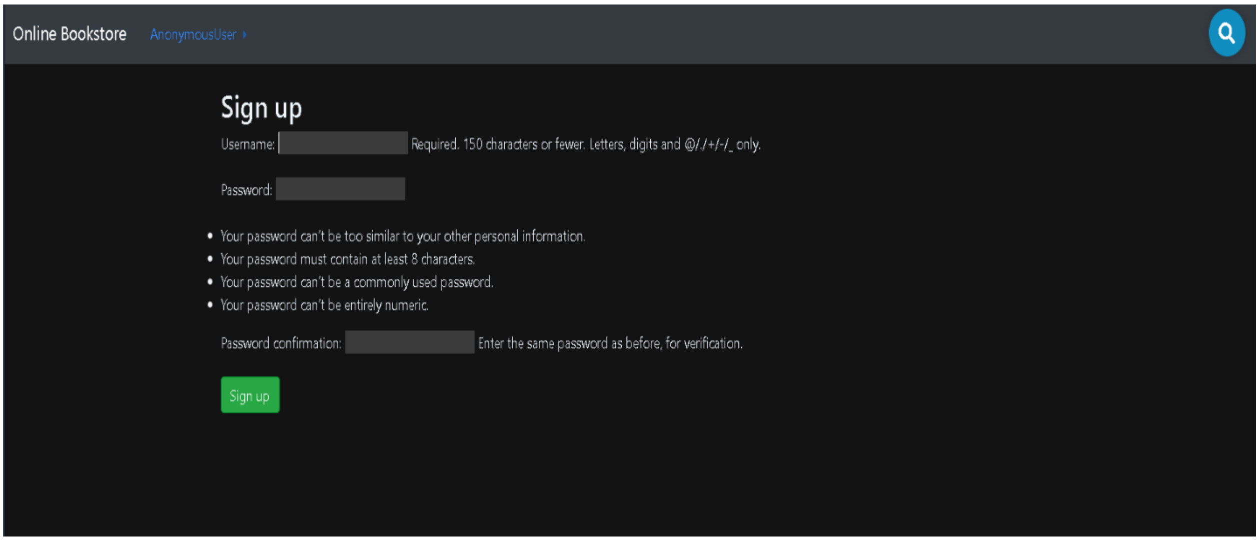
```
        paypal.Buttons({

            // Set up the transaction
            createOrder: function (data, actions) {
                return actions.order.create({
                    purchase_units: [{
                        amount: {
                            value: total
                        }
                    }]
                });
            },

            // Finalize the transaction
            onApprove: function (data, actions) {
                return actions.order.capture().then(function (details) {
                    // Show a success message to the buyer
                    completeOrder()
                    alert('Transaction completed by ' +
details.payer.name.given_name + '!');
                });
            }
        }).render('#paypal-button-container');
```

Chapter 6

SNAPSHOTS

6.1 The Login and Home Page

This is the users sign up page. All the users must sign up and once they are registered on our system they can login and only the authenticated users can purchase books from our system.



Sign up

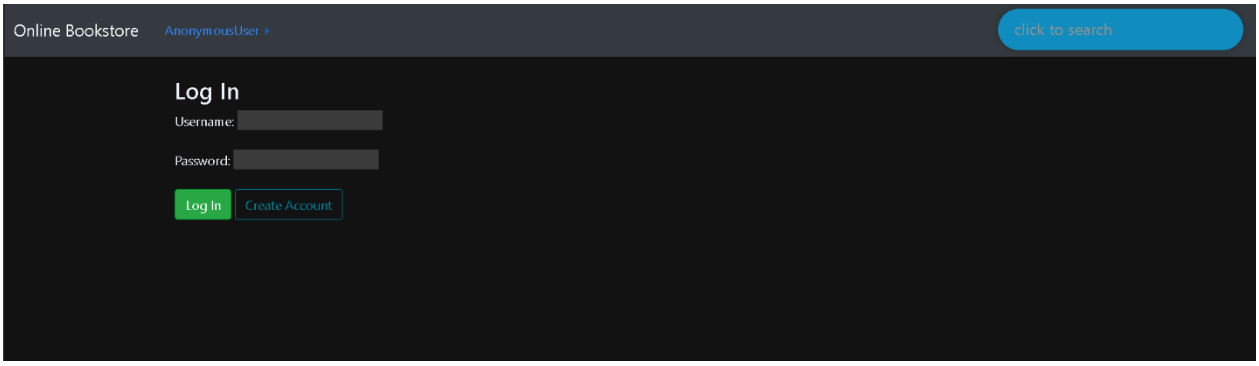
Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

Figure 6.1 Sign up page



Log In

Username:

Password:

Figure 6.2 Login Page

Figure 6.2 describes login page where only authenticated user can access our Book store by entering valid credentials the same will be scrutinized with the database via internal API.

6.2 The Home page

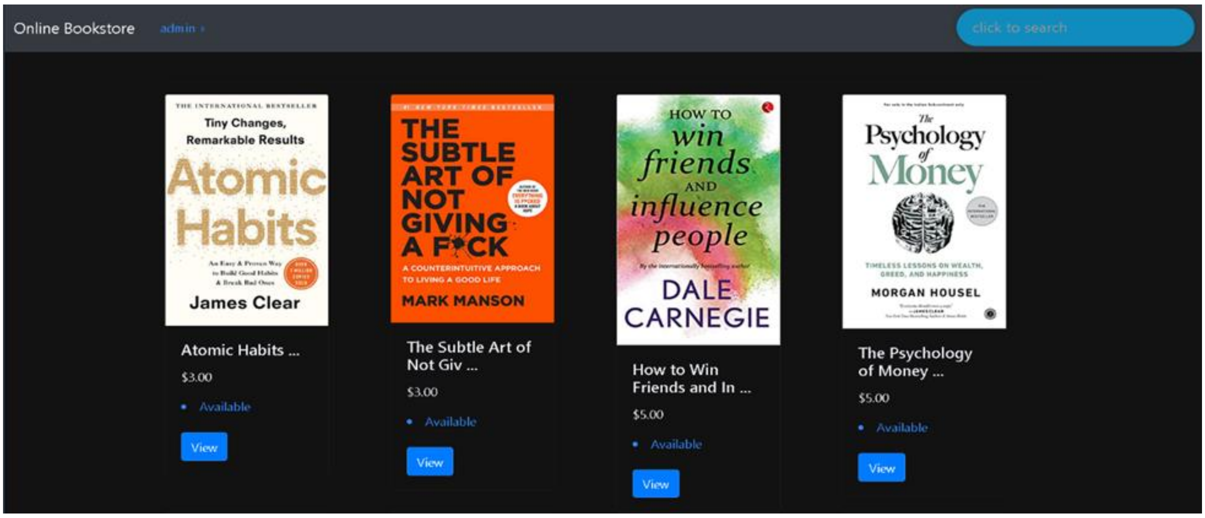
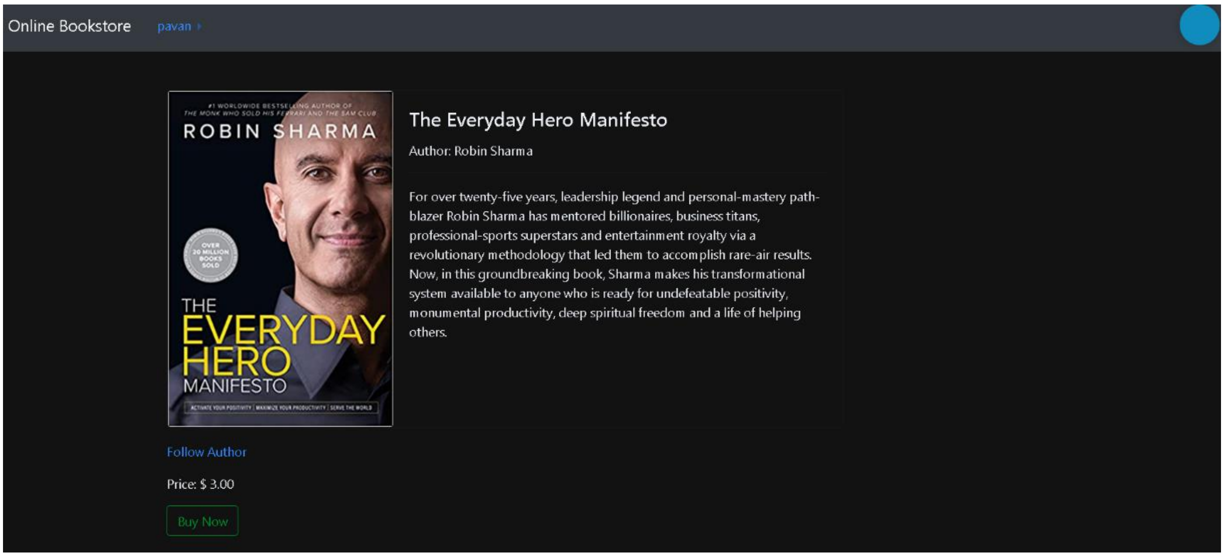


Figure 6.3 Home Page

6.3 Book Description Page

User can view the complete description of a book by clicking on the book card. All the details regarding that book will be fetched from our database and will be rendered on the UI. User can also add book to the cart and complete the payment process through paypal button and can check out the cart successfully.



The Everyday Hero Manifesto

Author: Robin Sharma

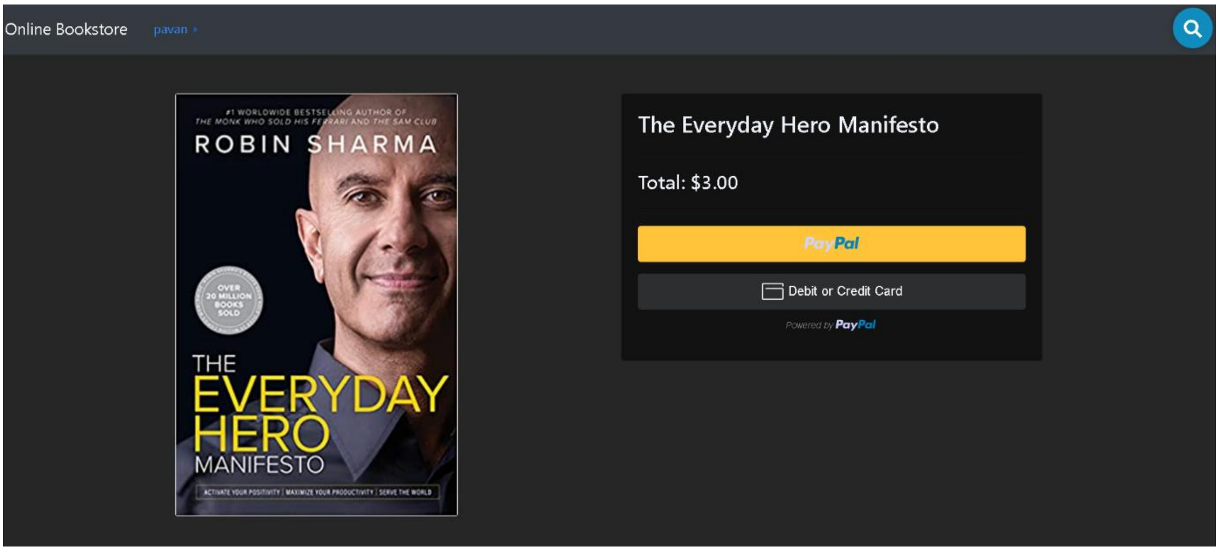
For over twenty-five years, leadership legend and personal-mastery pathblazer Robin Sharma has mentored billionaires, business titans, professional-sports superstars and entertainment royalty via a revolutionary methodology that led them to accomplish rare-air results. Now, in this groundbreaking book, Sharma makes his transformational system available to anyone who is ready for undefeatable positivity, monumental productivity, deep spiritual freedom and a life of helping others.

Price: $ 3.00

Figure 6.4 Book Description Page

6.4 Checkout Page

Figure 6.5 Checkout Page



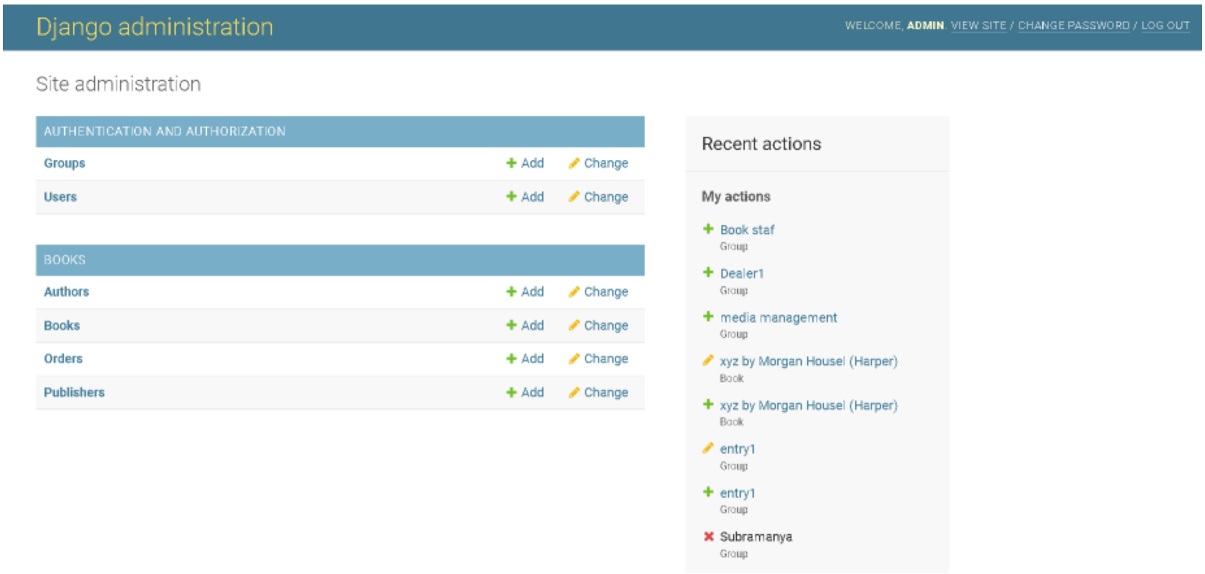
Online Bookstore pavan ›

The Everyday Hero Manifesto

Total: $3.00

PayPal
Debit or Credit Card
Powered by PayPal
#1 WORLDWIDE BESTSELLING AUTHOR OF THE MONK WHO SOLD HIS FERRARI AND THE 5AM CLUB
ROBIN SHARMA
OVER 20 MILLION BOOKS SOLD
THE EVERYDAY HERO MANIFESTO
ACTIVATE YOUR POSITIVITY | MAXIMIZE YOUR PRODUCTIVITY | SERVE THE WORLD

6.5 Admin Panel



Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

BOOKS		
Authors	+ Add	Change
Books	+ Add	Change
Orders	+ Add	Change
Publishers	+ Add	Change

Recent actions

My actions

- + Book staf
Group
- + Dealer1
Group
- + media management
Group
- ✎ xyz by Morgan Housel (Harper)
Book
- + xyz by Morgan Housel (Harper)
Book
- ✎ entry1
Group
- + entry1
Group
- ✖ Subramanya
Group

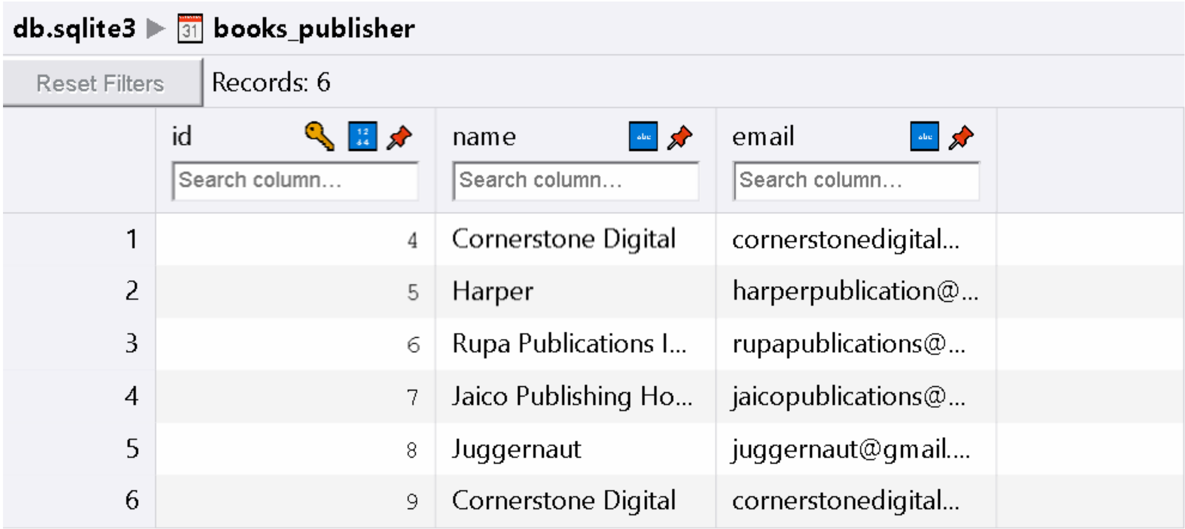
Figure 6.6 Admin Panel

Admin will use this application to give them access to the person who wants to purchase books and make a contribution by selling old books, the even admin will add new data regarding books so that the user can easily access it and purchase the books on demand.

6.6 Table Structure

➤ Publisher Table

Table Name: books_publisher



db.sqlite3 ▶ books_publisher

Reset Filters Records: 6

	id	name	email
	Search column...	Search column...	Search column...
1	4	Cornerstone Digital	cornerstonedigital...
2	5	Harper	harperpublication@...
3	6	Rupa Publications I...	rupapublications@...
4	7	Jaico Publishing Ho...	jaicopublications@...
5	8	Juggernaut	juggernaut@gmail....
6	9	Cornerstone Digital	cornerstonedigital...

Figure 6.7 Publisher Table

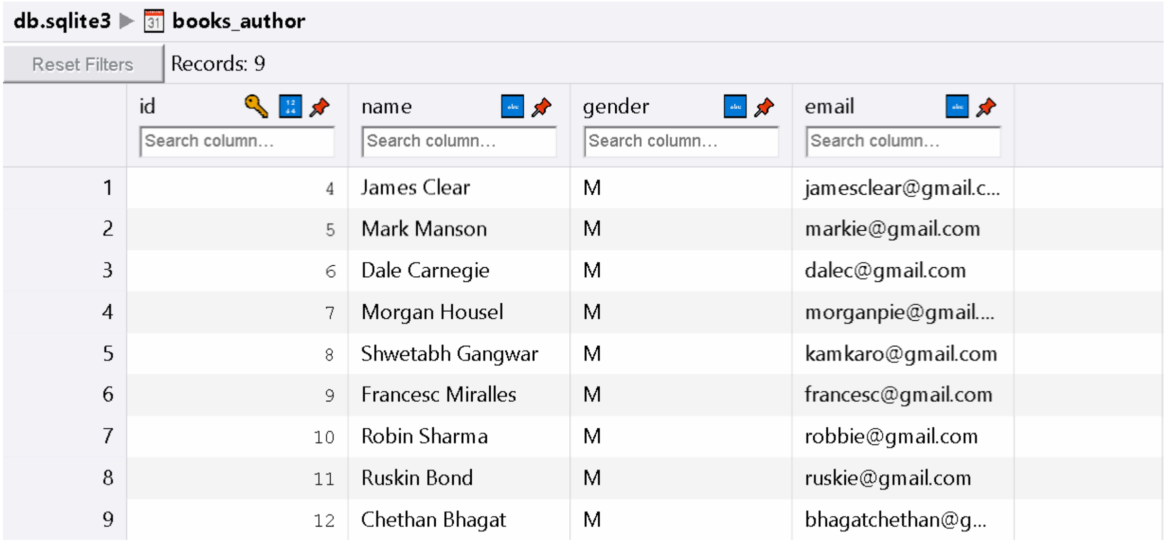
Table 6.1 Publisher Table

Fieldname	**Datatype**	**Length**	**Key**
Id	Integer	11	Primary
Name	Varchar	200	
Email	Varchar	100	

Here Id is a primary key. Name and Email are other attributes of publisher table.

➢ Author Table

Table Name: books_author



db.sqlite3 ▶ books_author

Reset Filters Records: 9

	id	name	gender	email
	Search column...	Search column...	Search column...	Search column...
1	4	James Clear	M	jamesclear@gmail.c...
2	5	Mark Manson	M	markie@gmail.com
3	6	Dale Carnegie	M	dalec@gmail.com
4	7	Morgan Housel	M	morganpie@gmail....
5	8	Shwetabh Gangwar	M	kamkaro@gmail.com
6	9	Francesc Miralles	M	francesc@gmail.com
7	10	Robin Sharma	M	robbie@gmail.com
8	11	Ruskin Bond	M	ruskie@gmail.com
9	12	Chethan Bhagat	M	bhagatchethan@g...

Figure 6.8 Authors Table

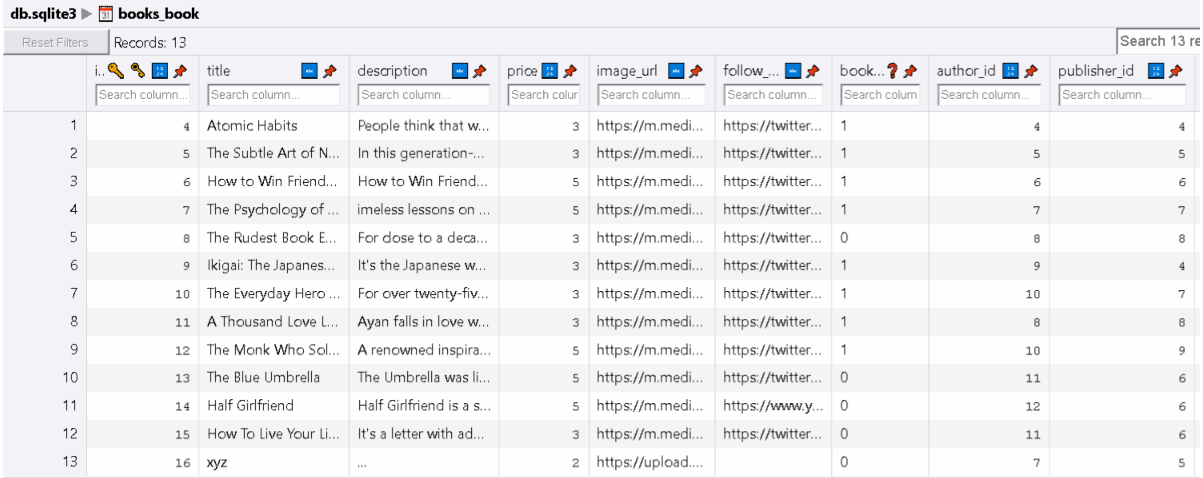
Table 6.2 Author Table

Fieldname	**Datatype**	**Length**	**Key**
Id	Integer	11	Primary
Name	Varchar	200	
Gender	Char	1	
Email	Varchar	200	

Here Id is a primary key. Name, Gender and Email are other attributes of Author table.

➢ Book Table

Table Name: books_book



db.sqlite3 ▶ books_book

Reset Filters | Records: 13 | Search 13 re

	i..	title	description	price	image_url	follow_...	book...	author_id	publisher_id
1	4	Atomic Habits	People think that w...	3	https://m.medi...	https://twitter...	1	4	4
2	5	The Subtle Art of N...	In this generation-...	3	https://m.medi...	https://twitter...	1	5	5
3	6	How to Win Friend...	How to Win Friend...	5	https://m.medi...	https://twitter...	1	6	6
4	7	The Psychology of ...	imeless lessons on ...	5	https://m.medi...	https://twitter...	1	7	7
5	8	The Rudest Book E...	For dose to a deca...	3	https://m.medi...	https://twitter...	0	8	8
6	9	Ikigai: The Japanes...	It's the Japanese w...	3	https://m.medi...	https://twitter...	1	9	4
7	10	The Everyday Hero ...	For over twenty-fiv...	3	https://m.medi...	https://twitter...	1	10	7
8	11	A Thousand Love L...	Ayan falls in love w...	3	https://m.medi...	https://twitter...	1	8	8
9	12	The Monk Who Sol...	A renowned inspira...	5	https://m.medi...	https://twitter...	1	10	9
10	13	The Blue Umbrella	The Umbrella was li...	5	https://m.medi...	https://twitter...	0	11	6
11	14	Half Girlfriend	Half Girlfriend is a s...	5	https://m.medi...	https://www.y...	0	12	6
12	15	How To Live Your Li...	It's a letter with ad...	3	https://m.medi...	https://twitter...	0	11	6
13	16	xyz	...	2	https://upload....		0	7	5

Figure 6.9 Books Table

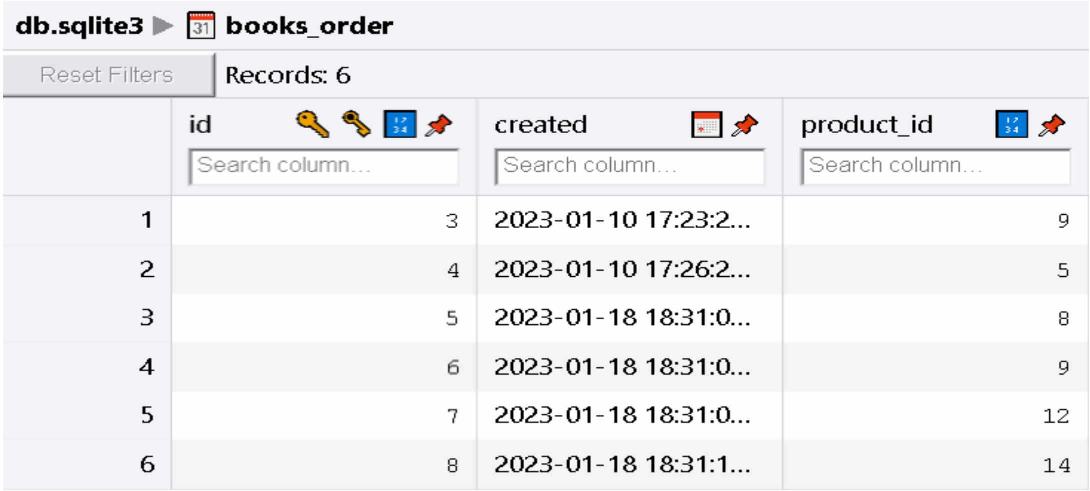
Table 6.3 Book Table

Fieldname	Datatype	Length	Key
Id	Integer	11	Primary
Title	Varchar	200	
Author	Varchar	200	Reference
Publisher	Varchar	200	Reference
Description	Varchar	500	
Price	Float	10	
Image_url	Varchar	2083	
Follow_author	Varchar	2083	
Book_available	Boolean		

Here id is a primary key. Title, description, price, image url, follow author and book available are other attributes of publisher table. Author and publisher are foreign key referencing.

➢ Order Table

Table Name: books_order



db.sqlite3 ▶ books_order

Records: 6

	id	created	product_id
1	3	2023-01-10 17:23:2...	9
2	4	2023-01-10 17:26:2...	5
3	5	2023-01-18 18:31:0...	8
4	6	2023-01-18 18:31:0...	9
5	7	2023-01-18 18:31:0...	12
6	8	2023-01-18 18:31:1...	14

Figure 6.10 Orders Table

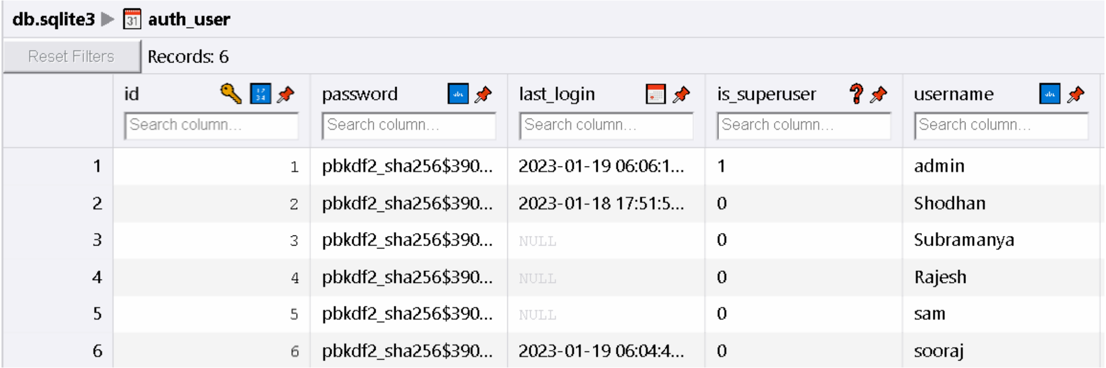
Table 6.4 Order Table

Fieldname	**Datatype**	**Length**	**Key**
id	Integer	11	Primary
product	Varchar	100	Reference
created	Date		

Here Id is the primary key. Product is a Foreign key referencing book table.

➢ User Table

Table Name: auth_user



	id	password	last_login	is_superuser	username
1	1	pbkdf2_sha256$390...	2023-01-19 06:06:1...	1	admin
2	2	pbkdf2_sha256$390...	2023-01-18 17:51:5...	0	Shodhan
3	3	pbkdf2_sha256$390...	NULL	0	Subramanya
4	4	pbkdf2_sha256$390...	NULL	0	Rajesh
5	5	pbkdf2_sha256$390...	NULL	0	sam
6	6	pbkdf2_sha256$390...	2023-01-19 06:04:4...	0	sooraj

Figure 6.11 Users Table

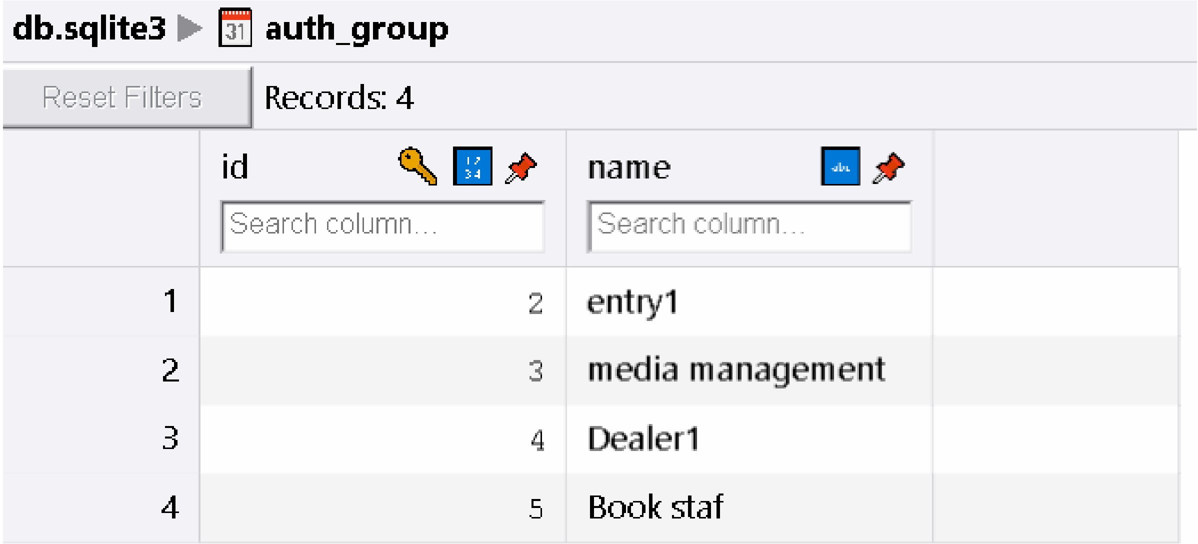
Table 6.5 User Table

Fieldname	**Datatype**	**Length**	**Key**
Id	Integer	11	Primary
Password	Varchar	50	
Last_login	Date		
Is_superuser	boolean	1	
Username	varchar	20	
Last_name	Varchar	20	
Email	Varchar	50	
Is_staff	Boolean		
Is_active	Boolean		
Date_joined	Date		
First_name	Varchar	20	

Here Id is the primary key. Password, last login, username, last name, email, first name, date joined etc are other attributes of User table

➢ Group Table

Table Name: auth_group



db.sqlite3 ▶ auth_group

Reset Filters | Records: 4

	id	name
1	2	entry1
2	3	media management
3	4	Dealer1
4	5	Book staf

Figure 6.12 Group Table

Table 6.6 Group Table

Fieldname	**Datatype**	**Length**	**Key**
Id	Integer	11	Primary
Name	Varchar	50	

Here Id is the primary key. Name is the attribute of Group table.

Chapter 7

CONCLUSION

The switch from written books being from bookstores to being ordered online or even just digital copies has had profound effects on the industry including bookstores and libraries and the general people of the world. The positives include easy access for everyone and cheaper books along with saving natural resources. The advantages of implementing the Online book store is to the present systems used by the library. The most admirable feature found 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.

FUTURE ENHANCEMENTS

Our project is a fairly simple Database consisting of a few fields, adding more information and being able to manage more student information with added functionality.Further we can also categorize the books based on various Genre's so that user can easily filter out books accordingly.We can also implement a star based review system for each book so that a user can buy books based on their reviews.User Interface and User Experience can be further enhanced by using advanced frontend frameworks rather than server-side rendering the templatesHowever, there are many scopes to modify this system. Due to shortage of time, we here unable to include many features

REFERENCES

1. Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition,2017,Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.
3. The Bootstrap Website (for Design) – https://getbootstrap.com/
4. W3 Schools (HTML, CSS and PHP reference) –https://www.w3schools.com/html
5. The SQLite Documentation – https://www.sqlite.org/docs.html
6. Django Official Documentation – https://www.djangoproject.com/
7. Stack Overflow – https://stackoverflow.com/
8. Django templates – https://docs.djangoproject.com/en/4.1/ref/templates/api/#django.template.Engine
9. Paypal Dashboard – https://developer.paypal.com/dashboard/
10. Project Repository – https://github.com/shodhanshetty14/BookStore-Django