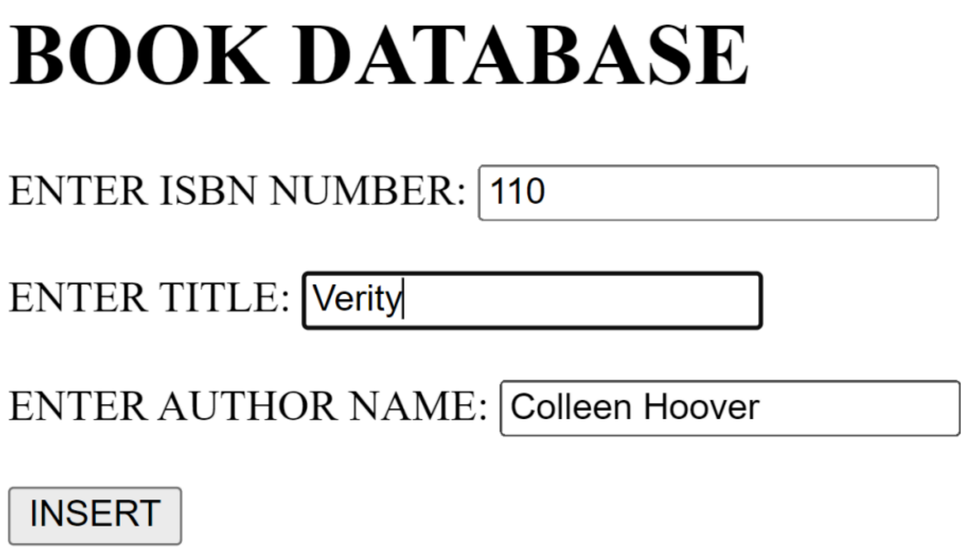
LAB TASKS:
Task-01: Create a table "Books" with fields like ISBN_NO, Title, and Author.
Create an html form to insert data into using PHP., and display the data using an html table.

HTML FORM:



BOOK DATABASE

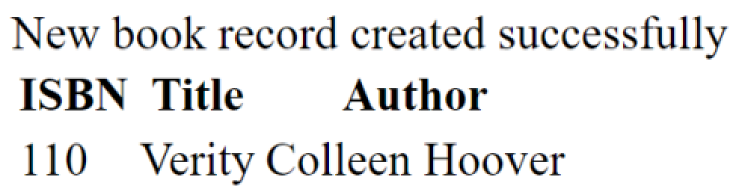
ENTER ISBN NUMBER: 110

ENTER TITLE: Verity

ENTER AUTHOR NAME: Colleen Hoover

INSERT

AFTER PRESSING INSERT BUTTON: (DISPLAY EXISTING RECORDS):



New book record created successfully

ISBN	Title	Author
110	Verity	Colleen Hoover

SOURCE CODE:

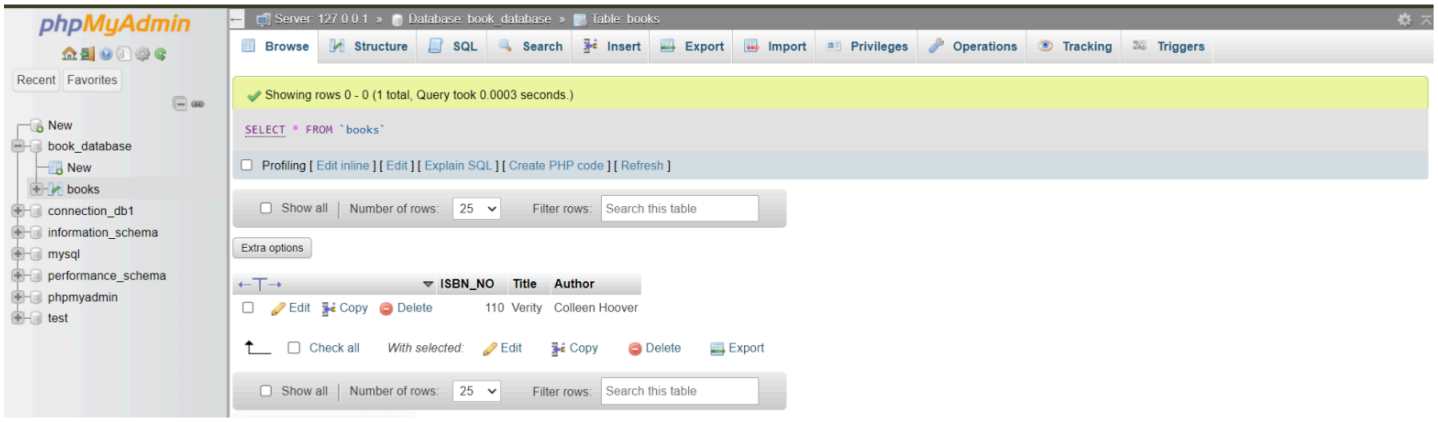
Index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Book Database</title>
</head>
<body>
    <h1>BOOK DATABASE</h1>
    <form action="connect.php" method="post">
        <label for="isbn">ENTER ISBN NUMBER:</label>
        <input type="number" id="isbn" name="isbn" required><br><br>
        <label for="title">ENTER TITLE:</label>
        <input type="text" id="title" name="title" required><br><br>
        <label for="author">ENTER AUTHOR NAME:</label>
        <input type="text" id="author" name="author" required><br><br>
        <button type="submit">INSERT</button>
    </form>
</body>
</html>
```

Connect.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "book_database";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $ISBN_NUM = $_POST["isbn"];
    $TITLE = $_POST["title"];
    $AUTHOR = $_POST["author"];
    $sql = "INSERT INTO books (ISBN_NO, Title, Author) VALUES (?, ?, ?)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("sss", $ISBN_NUM, $TITLE, $AUTHOR);
    $stmt->execute();
    if ($stmt->affected_rows > 0) {
        echo "New book record created successfully";
    } else {
        echo "Error: " . $stmt->error;
    }
}

// Display the Existing Records in the Book Table
$sql = "SELECT * FROM Books";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<table>";
    echo "<tr><th>ISBN</th><th>Title</th><th>Author</th></tr>";
    while ($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["ISBN_NO"] . "</td><td>" . $row["Title"] . "</td><td>" .
$row["Author"] . "</td></tr>";
    }
    echo "</table>";
} else {
    echo "NO BOOKS FOUND";
}

$conn->close();
?>
```

DATABASE:



Server: 127.0.0.1 » Database: book_database » Table: books

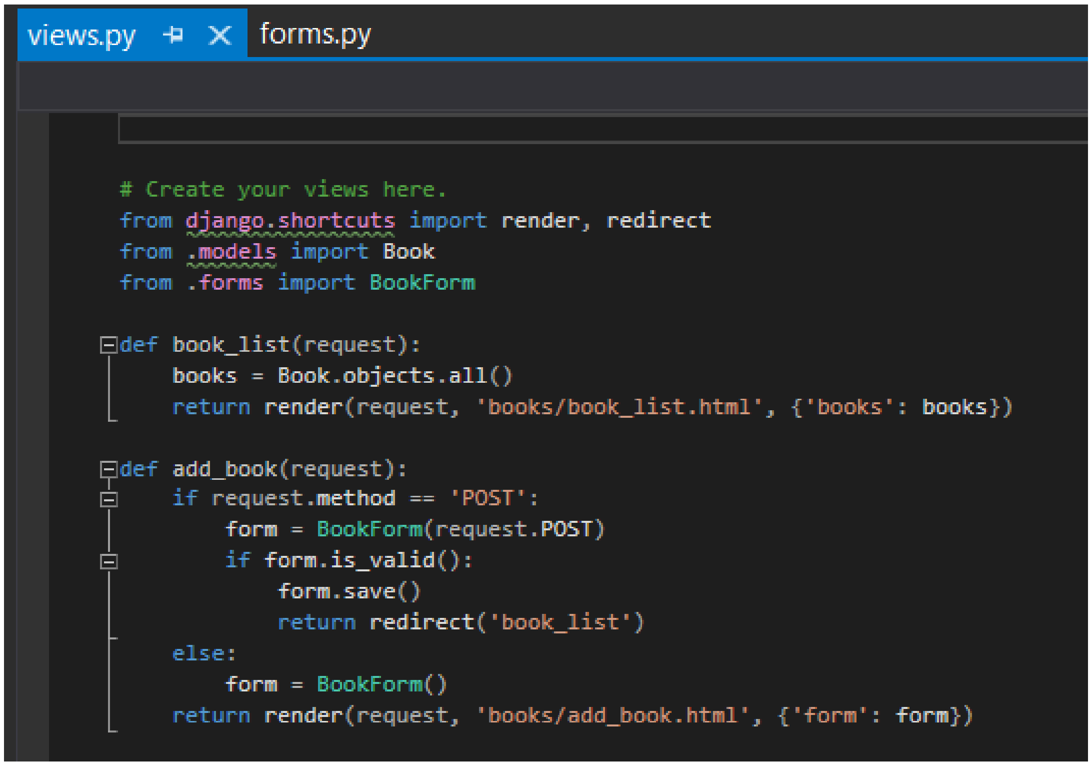
Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT * FROM `books`
```

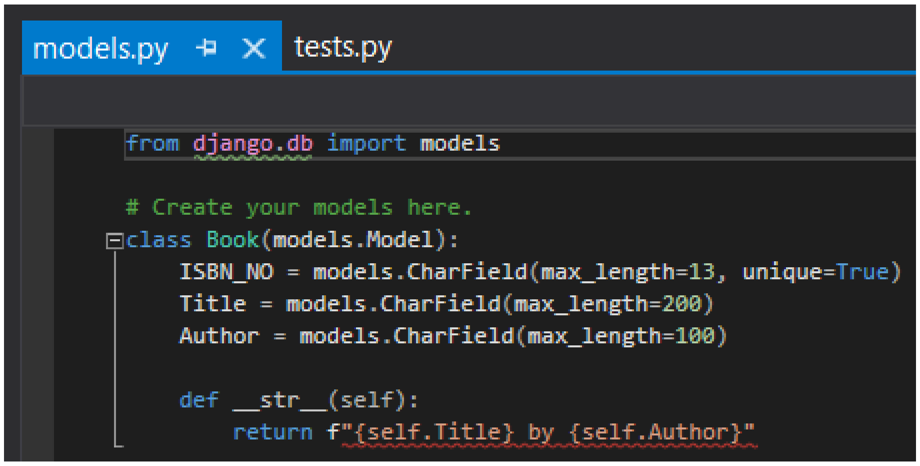
ISBN_NO	Title	Author
110	Verity	Colleen Hoover

Task-02: Replicate the same using C# with SQL Server using Windows Form. ***(note: done in python)***

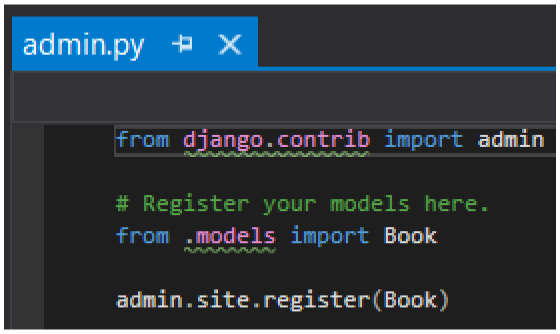
SOURCE CODES:



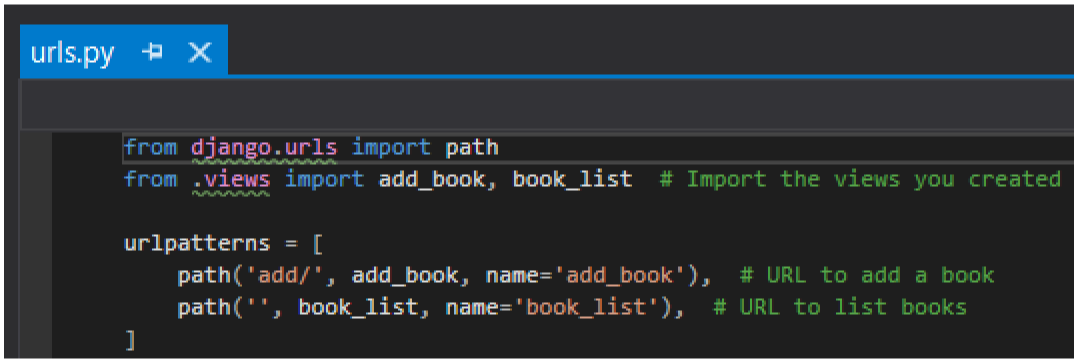
```
# Create your views here.
from django.shortcuts import render, redirect
from .models import Book
from .forms import BookForm

def book_list(request):
    books = Book.objects.all()
    return render(request, 'books/book_list.html', {'books': books})

def add_book(request):
    if request.method == 'POST':
        form = BookForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('book_list')
    else:
        form = BookForm()
    return render(request, 'books/add_book.html', {'form': form})
```



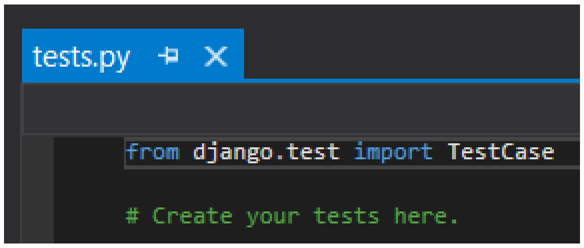
```
from django.db import models

# Create your models here.
class Book(models.Model):
    ISBN_NO = models.CharField(max_length=13, unique=True)
    Title = models.CharField(max_length=200)
    Author = models.CharField(max_length=100)

    def __str__(self):
        return f"{self.Title} by {self.Author}"
```



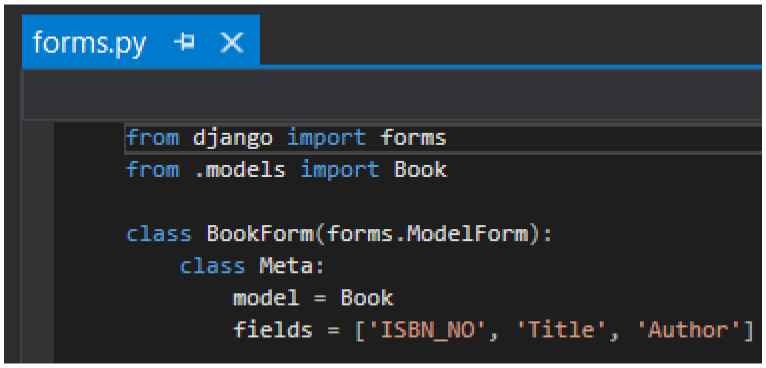
```
from django.contrib import admin

# Register your models here.
from .models import Book

admin.site.register(Book)
```



```python
from django.urls import path
from .views import add_book, book_list  # Import the views you created

urlpatterns = [
    path('add/', add_book, name='add_book'),  # URL to add a book
    path('', book_list, name='book_list'),  # URL to list books
]
```

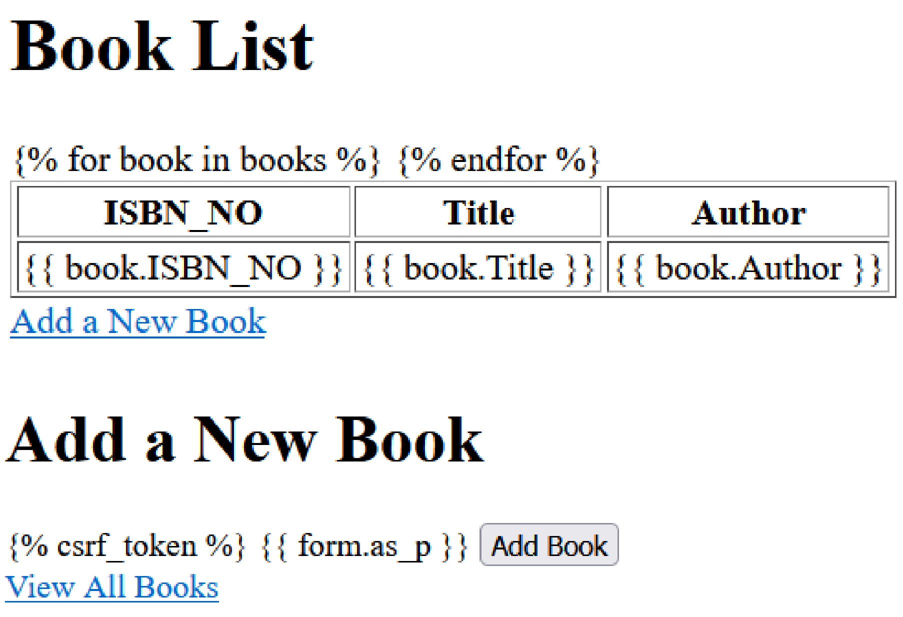


```python
from django.test import TestCase

# Create your tests here.
```



```python
from django import forms
from .models import Book

class BookForm(forms.ModelForm):
    class Meta:
        model = Book
        fields = ['ISBN_NO', 'Title', 'Author']
```

HTML FORM:



Book List

{% for book in books %} {% endfor %}

ISBN_NO	Title	Author
{{ book.ISBN_NO }}	{{ book.Title }}	{{ book.Author }}

Add a New Book

Add a New Book

{% csrf_token %} {{ form.as_p }} Add Book

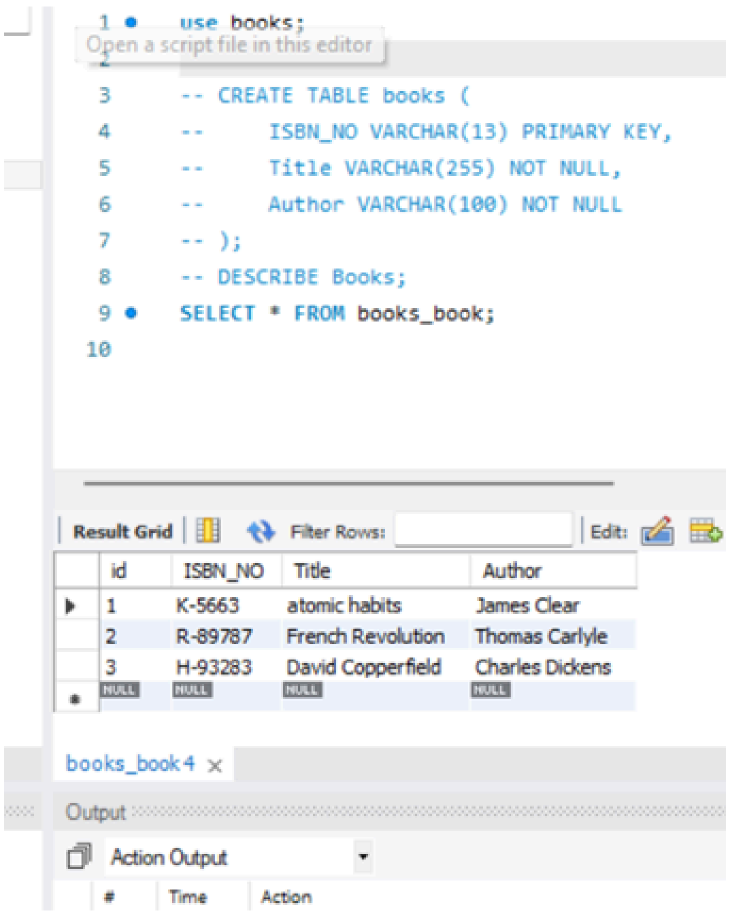
View All Books

Task-03: Perform the above task with Java using MYSQL in Visual Studio Code.

<u>Main.java (with MySQL Connection)</u>

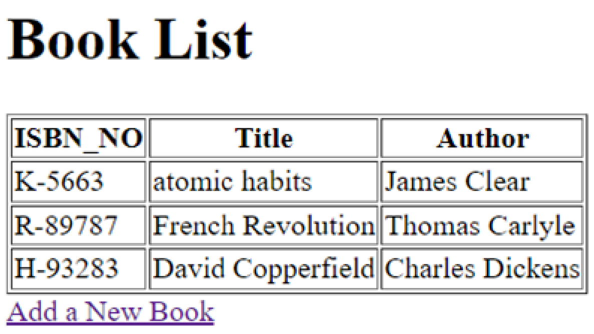
```
import java.sql.*;
import java.util.Scanner;

public class Main {
   // MySQL connection parameters
   static final String DB_URL = "jdbc:mysql://localhost:3306/bookdb"; // Update
your DB name and port if necessary
   static final String USER = "your_username";
   static final String PASS = "your_password";

   public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
       try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS))
{
           System.out.println("Connected to the database!");

           int choice;
           do {
               System.out.println("1. Add Book");
               System.out.println("2. View Books");
               System.out.println("3. Update Book");
               System.out.println("4. Delete Book");
               System.out.println("5. Exit");
               System.out.print("Enter your choice: ");
               choice = sc.nextInt();
               sc.nextLine(); // Consume newline

               switch (choice) {
                   case 1:
                       addBook(conn, sc);
                       break;
                   case 2:
                       viewBooks(conn);
                       break;
                   case 3:
                       updateBook(conn, sc);
                       break;
                   case 4:
                       deleteBook(conn, sc);
                       break;
                   case 5:
                       System.out.println("Exiting...");
                       break;
                   default:
                       System.out.println("Invalid choice.");
               }
```

```
            } while (choice != 5);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    // Add a new book
    public static void addBook(Connection conn, Scanner sc) throws SQLException
{
        System.out.print("Enter book ISBN: ");
        String isbn = sc.nextLine();
        System.out.print("Enter book title: ");
        String title = sc.nextLine();
        System.out.print("Enter book author: ");
        String author = sc.nextLine();

        String query = "INSERT INTO books (isbn, title, author) VALUES (?, ?,
?)";
        try (PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setString(1, isbn);
            pstmt.setString(2, title);
            pstmt.setString(3, author);
            pstmt.executeUpdate();
            System.out.println("Book added successfully.");
        }
    }

    // View all books
    public static void viewBooks(Connection conn) throws SQLException {
        String query = "SELECT * FROM books";
        try (Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(query)) {

            System.out.println("Books:");
            System.out.println("ID | ISBN | Title | Author");
            while (rs.next()) {
                System.out.println(rs.getInt("id") + " | " +
rs.getString("isbn") + " | " + rs.getString("title") + " | " +
rs.getString("author"));
            }
        }
    }

    // Update a book's details
    public static void updateBook(Connection conn, Scanner sc) throws
SQLException {
        System.out.print("Enter book ID to update: ");
        int id = sc.nextInt();
        sc.nextLine(); // Consume newline
        System.out.print("Enter new ISBN: ");
        String newIsbn = sc.nextLine();
        System.out.print("Enter new title: ");
        String newTitle = sc.nextLine();
        System.out.print("Enter new author: ");
        String newAuthor = sc.nextLine();

        String query = "UPDATE books SET isbn = ?, title = ?, author = ? WHERE
id = ?";
        try (PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setString(1, newIsbn);
            pstmt.setString(2, newTitle);
            pstmt.setString(3, newAuthor);
            pstmt.setInt(4, id);
            int rowsAffected = pstmt.executeUpdate();
            if (rowsAffected > 0) {
                System.out.println("Book updated successfully.");
            } else {
                System.out.println("Book not found.");
            }
        }
    }

    // Delete a book by ID
    public static void deleteBook(Connection conn, Scanner sc) throws
SQLException {
        System.out.print("Enter book ID to delete: ");
        int id = sc.nextInt();

        String query = "DELETE FROM books WHERE id = ?";
        try (PreparedStatement pstmt = conn.prepareStatement(query)) {
            pstmt.setInt(1, id);
            int rowsAffected = pstmt.executeUpdate();
        }
    }
}
```



```
use books;

-- CREATE TABLE books (
--     ISBN_NO VARCHAR(13) PRIMARY KEY,
--     Title VARCHAR(255) NOT NULL,
--     Author VARCHAR(100) NOT NULL
-- );
-- DESCRIBE Books;
SELECT * FROM books_book;
```

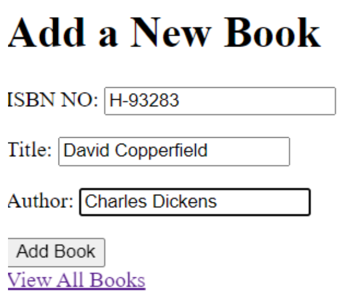
id	ISBN_NO	Title	Author
1	K-5663	atomic habits	James Clear
2	R-89787	French Revolution	Thomas Carlyle
3	H-93283	David Copperfield	Charles Dickens
NULL	NULL	NULL	NULL



Book List

ISBN_NO	Title	Author
K-5663	atomic habits	James Clear
R-89787	French Revolution	Thomas Carlyle
H-93283	David Copperfield	Charles Dickens

Add a New Book



Add a New Book

ISBN NO: H-93283

Title: David Copperfield

Author: Charles Dickens

View All Books