BOOK CATALOG

The Book Catalog System is a Python-based graphical user interface (GUI) application designed to manage a collection of books efficiently.

By,
Jissmol Jose
Sreesha M.K

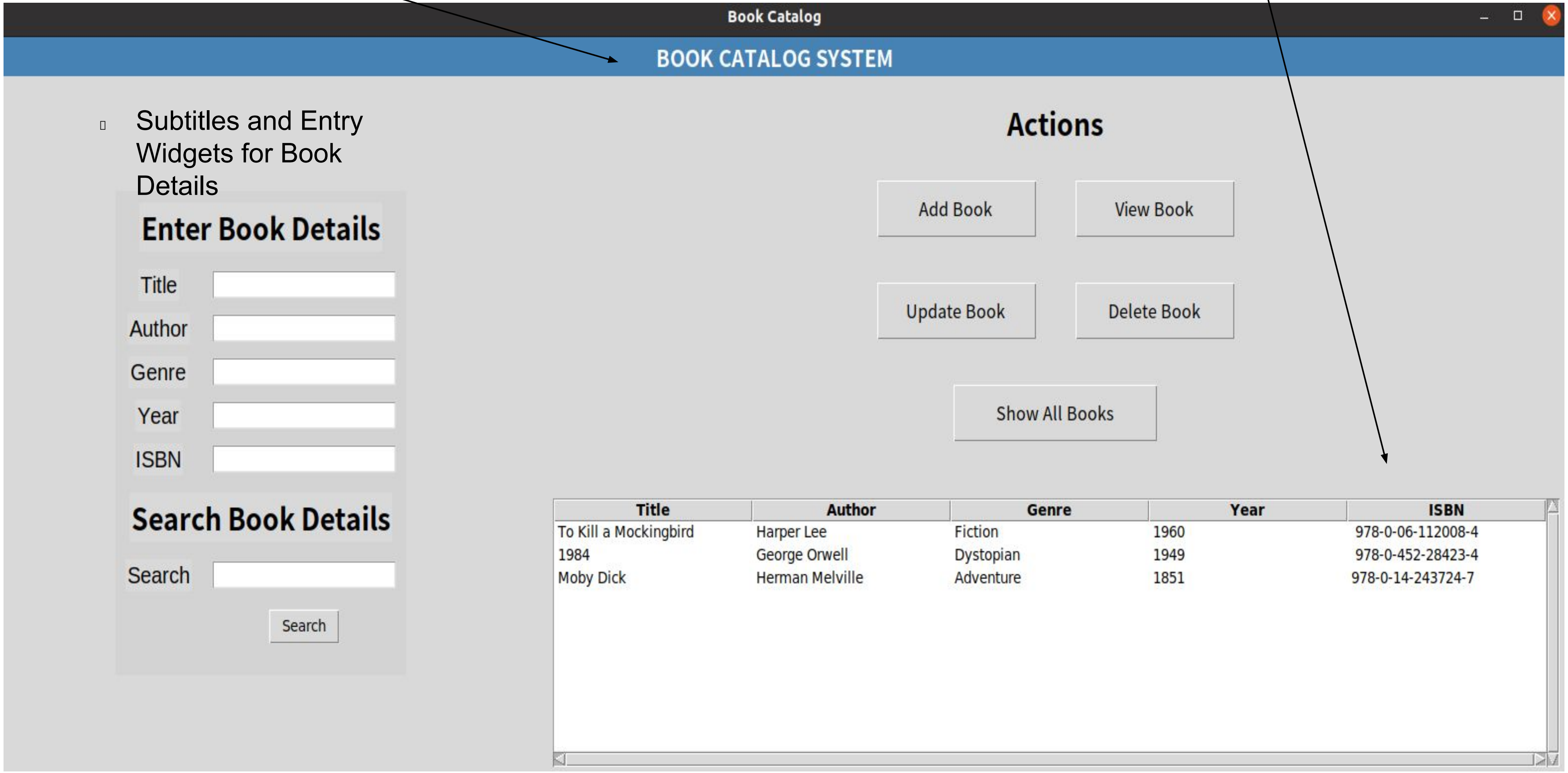
INTRODUCTION

- This project will help to practice Python skills
- This project aims to demonstrate the integration of a graphical user interface with a backend database system, showcasing the practical use of Tkinter and SQLite in building desktop applications.
- This project provides a user-friendly interface for adding, viewing, updating, searching, and deleting book records.

GUI

HEADER OF THE APPLICATION

TREEVIEW



Title	Author	Genre	Year	ISBN
To Kill a Mockingbird	Harper Lee	Fiction	1960	978-0-06-112008-4
1984	George Orwell	Dystopian	1949	978-0-452-28423-4
Moby Dick	Herman Melville	Adventure	1851	978-0-14-243724-7

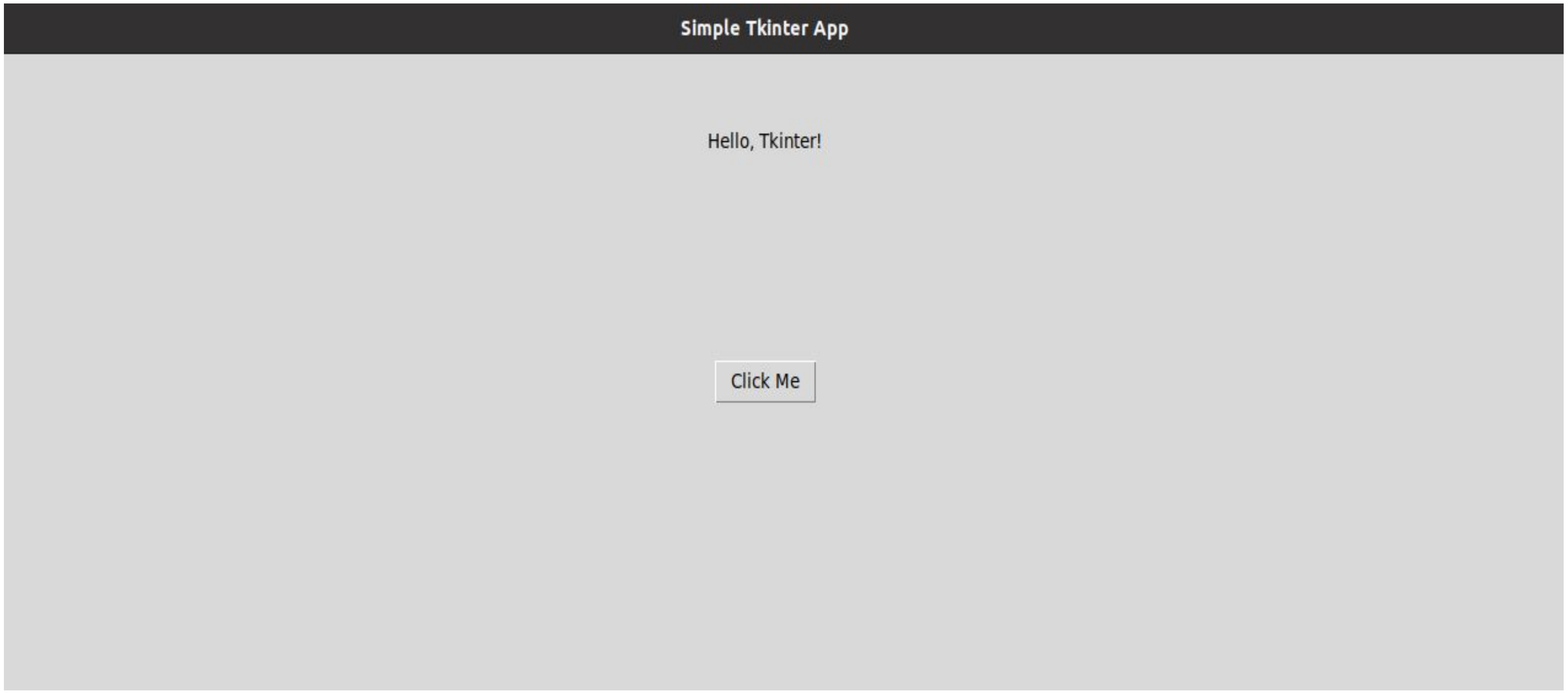
- Subtitles and Entry Widgets for Book Details

Basic Informations

- **IDE(integrated development environment)**
- Pycharm
- **Library's Used**
- Tkinter
- Sqlite3
- **Modules**
- Messagebox,simpledialog,ttk

TKINTER

- **Standard GUI (Graphical User Interface) library for Python.Creates the GUI for the application.**
- Key Components:
- Tk(): Creates the main window of the application.
- Label: Displays text or images.
- Entry: Provides text entry fields for user input.
- Button: Creates clickable buttons for user actions


Simple Tkinter App
Hello, Tkinter!
Click Me

WHAT IS SQULITE?

- # Sqlite3

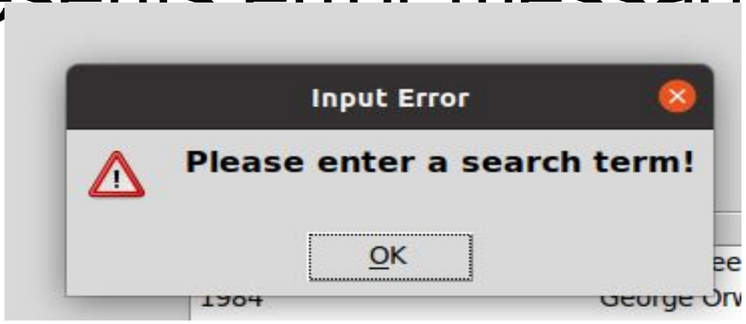
- **SQLite3 is a built-in Python library used for database management.Manages the SQLite database for storing book details.interface for interacting with SQLite databases, which are lightweight and serverless**
- Operations:
- Connect: Establishes a connection to the database file.
- Cursor: Executes SQL commands to interact with the database.

● ttk

- tkinter.ttk
- Provides themed widgets for a modern look.
- ttk stands for Tk themed widgets. It is part of the tkinter library
- Key Component:
- ttk.Treeview: Displays data in a table-like format.
- The tk button will have a classic, platform-dependent appearance.

MESSAGEBOX

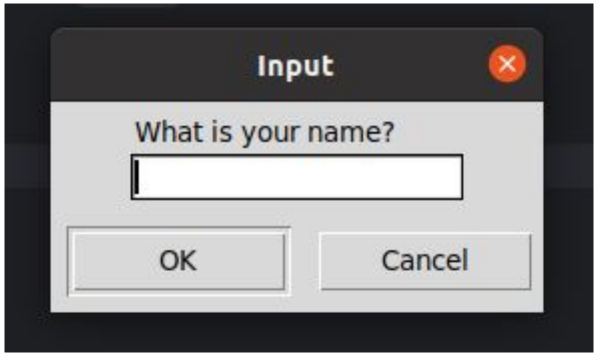
- Shows dialog boxes to communicate with the user.tkinter.messagebox
- Key Dialogs:
- showinfo: Displays informational messages.
- showwarning: Shows warning messages.
- showerror: Presents error messages.


Input Error

Please enter a search term!

SIMPLEDIALOG

- Provides simple dialogs for user input.
- Key Dialog:
   - askstring: Prompts the user for a string input.



Import Libraries and modules

- **SQLite library** to interact with SQLite databases.
- all classes and functions from the **tkinter module,** which is used for **creating GUI applications**
- **ttk module from tkinter** for more advanced widgets like **Treeview**.
- **messagebox** module for **displaying message dialogs.**
- **simpledialog** module for displaying **simple**

Database Setup

- **Connection:** Establishes a connection to an SQLite database file.If the file doesn't exist, it will be created.
- **Table Creation:**Creates a cursor object using the connection.This cursor is used to execute SQL queries.
- **SQL Commands:**connector.execute('SQL COMMAND', parameters).You can use this to create tables, insert data, update data, and delete data.
- **Transactions:** connector.commit(): Saves the

. Table Creation

- connector.execute(...): Executes an SQL statement to create a table named Books if it doesn't already exist.
- Title TEXT: Column for the book title (text type).
- Author TEXT: Column for the author's name (text type).
- Genre TEXT: Column for the book genre (text type).
- Year INTEGER: Column for the year of publication (integer type).

● FUNCTIONS

- **Clear Entries -clear_entries():**Clears all input fields in the GUI.clear_entries()
- **Add a Book -add_book():** Adds a new book to the database with details such as title, author, genre, publication year, and ISBN.
- **Display books- display_books():** Defines a function to display all books in the Treeview widget.
- **View All Books-book view_book()** Defines a

GRAPHICAL USER INTERFACE (GUI)

WHAT IS A GUI?

- A Graphical User Interface (GUI) is a user interface that allows users to interact with electronic devices using graphical elements such as windows, icons, buttons, and menus. GUIs are designed to be intuitive and user-friendly, providing a visual way to interact with a program without needing to use command-line commands.
- Event-Driven Programming: GUIs operate based on events. An event is an action or occurrence detected by the program (such as a mouse click, key press, or window resize). The GUI framework listens for these events and responds with predefined actions or functions, known as event handlers.
- Event Loop: GUIs rely on an event loop, which continuously checks for and dispatches events to the appropriate handlers. This loop allows the GUI to remain responsive to user input and other actions.
- Widgets and Components: GUIs are composed of various widgets or components (like buttons, text fields, etc.), which are often arranged in a hierarchical tree structure. Each widget has properties that define its behavior, appearance, and relationships with other widgets.

GUI SETUP

- Header: Displays the title of the application.
- Input Frame: Contains fields for entering book details and a search box.
- Button Frame: Includes buttons for various actions (Add, View, Update, Delete, Search, Show All).
- Treeview Frame: Shows the list of books in a table format with scrollbars.

Graphical user interface (GUI) for a book catalog system

- **<u>Initial Setup of the Tkinter Window</u>**

✔ Creates the main application window

✔ Sets the title of the window and dimension

✔ **<u>Header for the Application</u>**

✔ Sets the background color for the header.

✔ Label(...): Creates a label widget in the root window

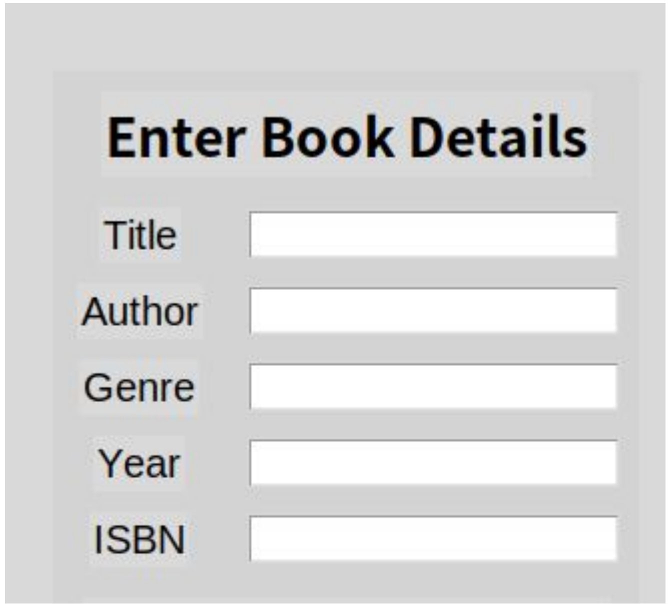
✔ pack(side=TOP, fill=X): Packs the label at the top of the window and stretches it horizontally

- **LEFT SIDE:-**

Frame for Input Fields
--input_frame = Frame(root): Creates a frame widget, which is a container for other widgets

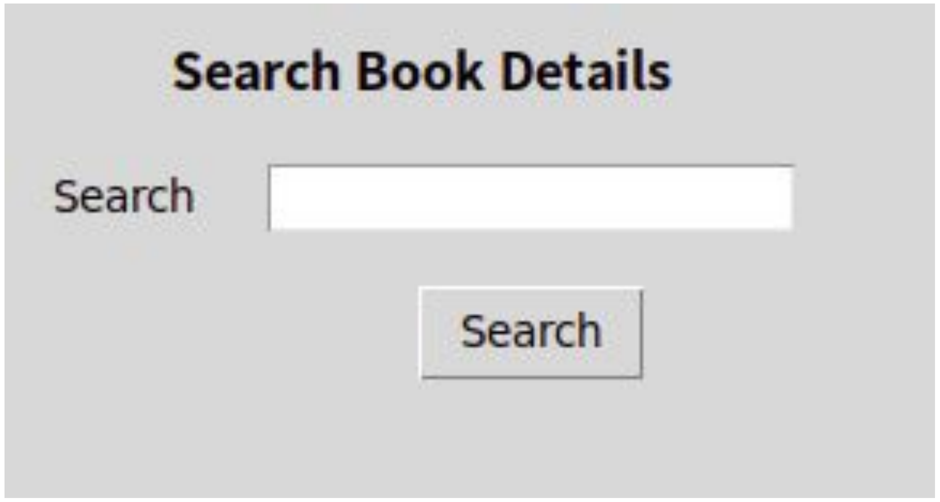
--frame.pack(side,padx,pady)
Packs the frame on the side you need with padding of pixels on the x and y axes

Subtitles and Entry Widgets for Book Details



- Label(...) and Entry(...): These lines create a label and an entry widget for entering the book's title.
- .grid(...): The label and entry widget are placed in the grid at specified row and column positions with padding.
- **Padding** refers to the space

Search Book Details Section



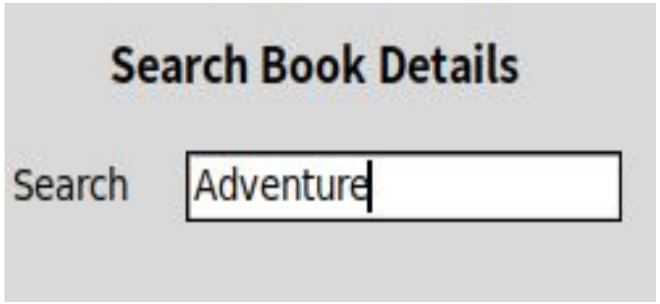
- Button(...): Each button is created with a label and a command function that gets executed when the button is clicked.
- Command here is search_book function

SEARCH BOOK

- function is designed to search for books in a database based on a user-provided search term.
- It searches across multiple fields (Title, Author, Genre, ISBN) and updates a Treeview widget with the result
- **<u>Retrieve and Validate Search Term</u>**
- retrieves the text
- mb.showwarning(), prompting the user to enter a valid search term

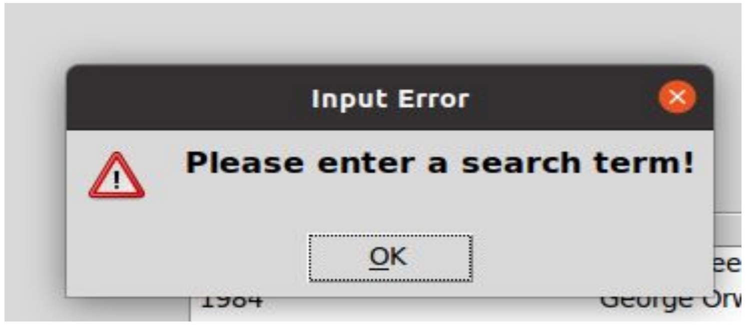
Execute the Search Query

- The SQL query searches the Books table for records where any of the fields (Title, Author, Genre, ISBN) contain the search term.
- LIKE with % wildcards allows partial matching, meaning any record containing the search term within these fields will be retrieved.
- connector.execute(query, (search_pattern, search_pattern, search_pattern, search_pattern)) safely executes the query with the search pattern, preventing SQL injection by



Search Book Details

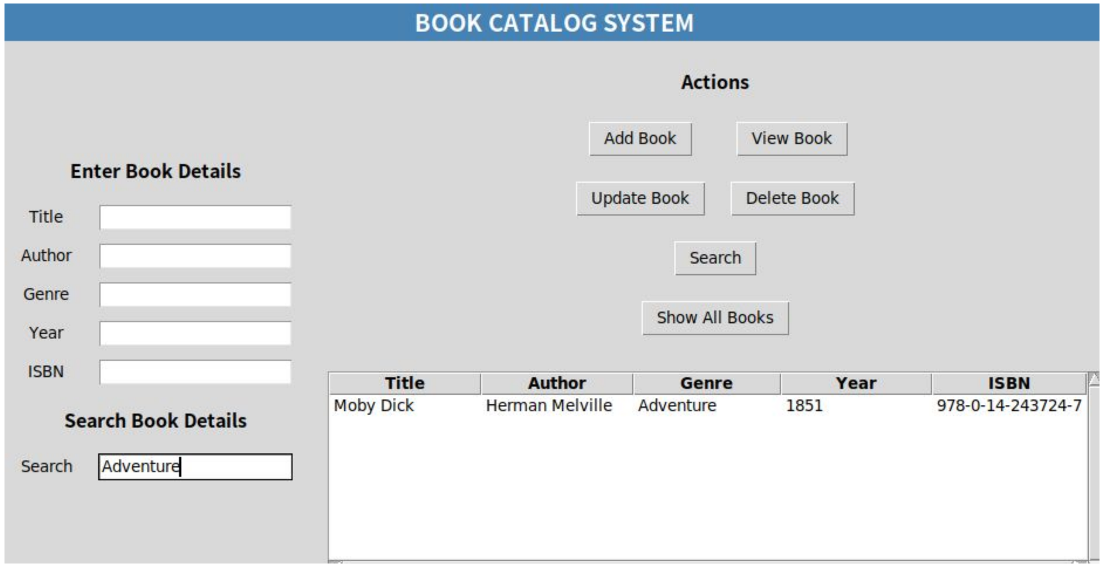
Search: Adventure



Input Error

Please enter a search term!

OK



BOOK CATALOG SYSTEM

Enter Book Details

Title:
Author:
Genre:
Year:
ISBN:

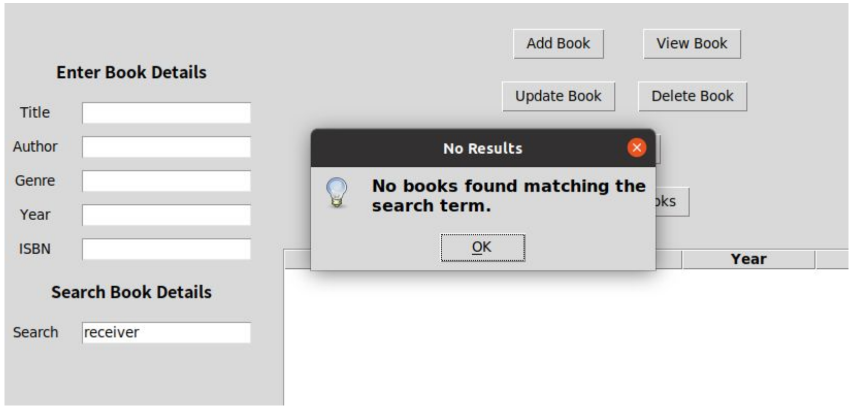
Search Book Details

Search: Adventure

Actions

Add Book | View Book | Update Book | Delete Book | Search | Show All Books

Title	Author	Genre	Year	ISBN
Moby Dick	Herman Melville	Adventure	1851	978-0-14-243724-7



Enter Book Details

Title:
Author:
Genre:
Year:
ISBN:

Search Book Details

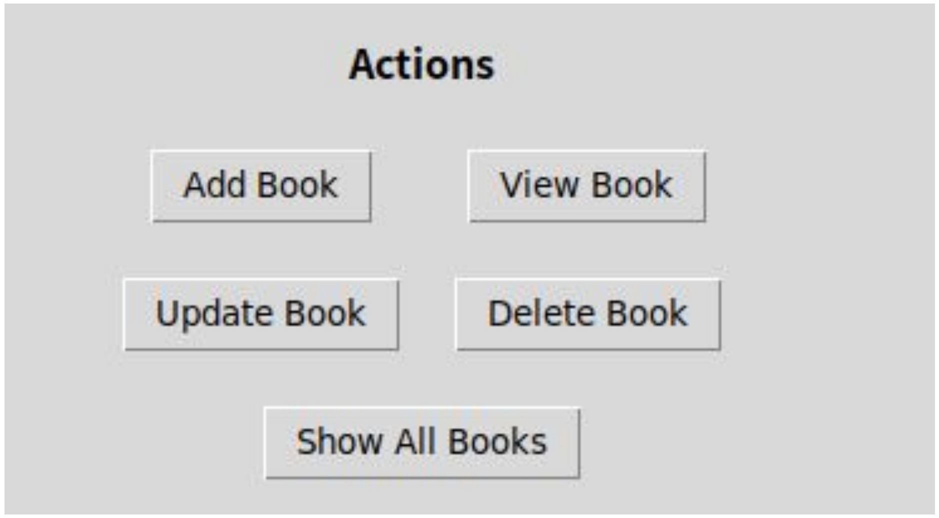
Search: receiver

Add Book | View Book | Update Book | Delete Book

No Results

No books found matching the search term.

OK



- button_frame: A frame is created to hold all the action buttons (e.g., "Add Book", "View Book", etc.).
- style = ttk.Style() creates a new style object.
- Button(...): Each

ADD BOOK

- **Retrieving Input Value**
- entry_field.get(): Retrieves the text from the entry_field.eg :field=title
- title:Pride and prejudice
- Validating Input
- **Checks if any of the input fields are empty.**
- mb.showwarning()-Displays a warning message if any fields are empty.return: Exits the function if any field is empty, preventing further execution

DISPLAY BOOK

- **Clearing the Treeview**
- tree: This is an instance of the Treeview widget used to display the list of books.
- tree.delete(*tree.get_children()): tree.get_children() returns a list of all items (rows) currently in the Treeview, and tree.delete(*...)
- **Querying the Database**
- Executes an SQL SELECT command to

- tree.insert('', END, values=row): Inserts each row into the Treeview widget.
- '': Specifies that the row will be inserted at the top level of the Treeview (not as a child of any other row).
- END: Specifies that the row will be inserted at the end of the current rows in the Treeview.
- values=row: Passes the tuple row as the data to be displayed in the new row of the Treeview.

Clearing the Input Fields

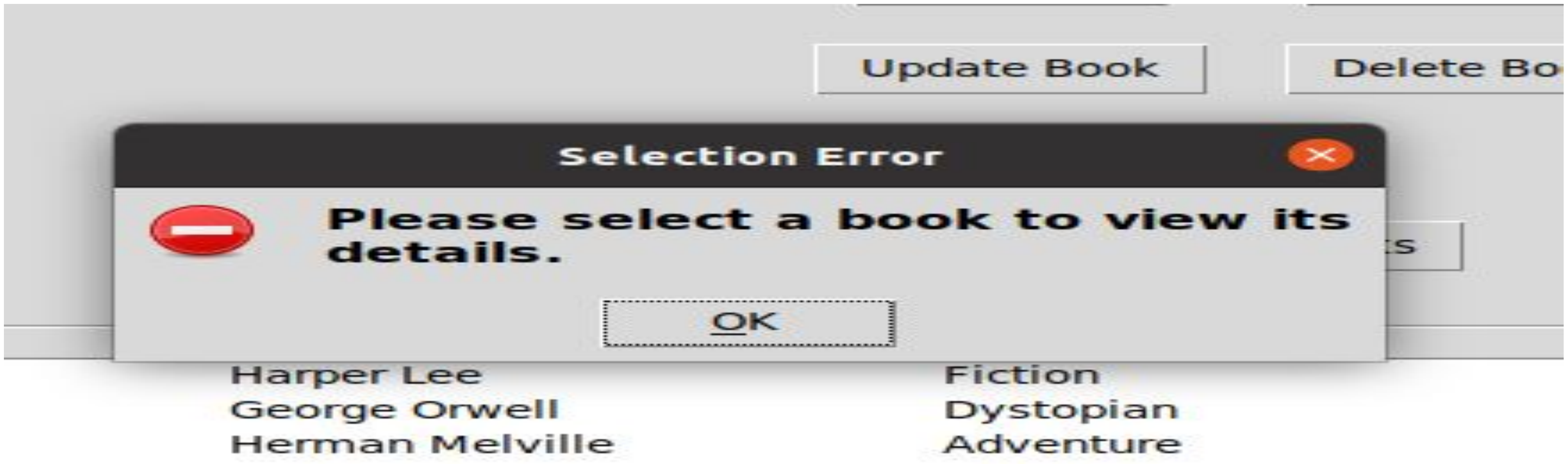
- **entry_title:** This is an instance of the Entry widget for the Inputs
- **delete(0, END):** This method clears the text in the entry_title widget. 0 is the starting index, and END represents the end of the text.
- The purpose of this function is to reset or clear the text fields in the GUI. It's typically used after an action such as adding, updating, or deleting a book, to ensure that the input fields are empty and ready for new data.

UPDATE BOOK

- The update_book function is used to update the details of a selected book in the Treeview widget and the database.
- **<u>Retrieving Selected Book Information</u>**
- isbn = selected_book[4]: Extracts the ISBN of the selected book, which is used to identify the record to update.
- **<u>Prompting User for New Details</u>**
- askstring()-prompts the user to enter new details for each field from tkinter.simpledialog

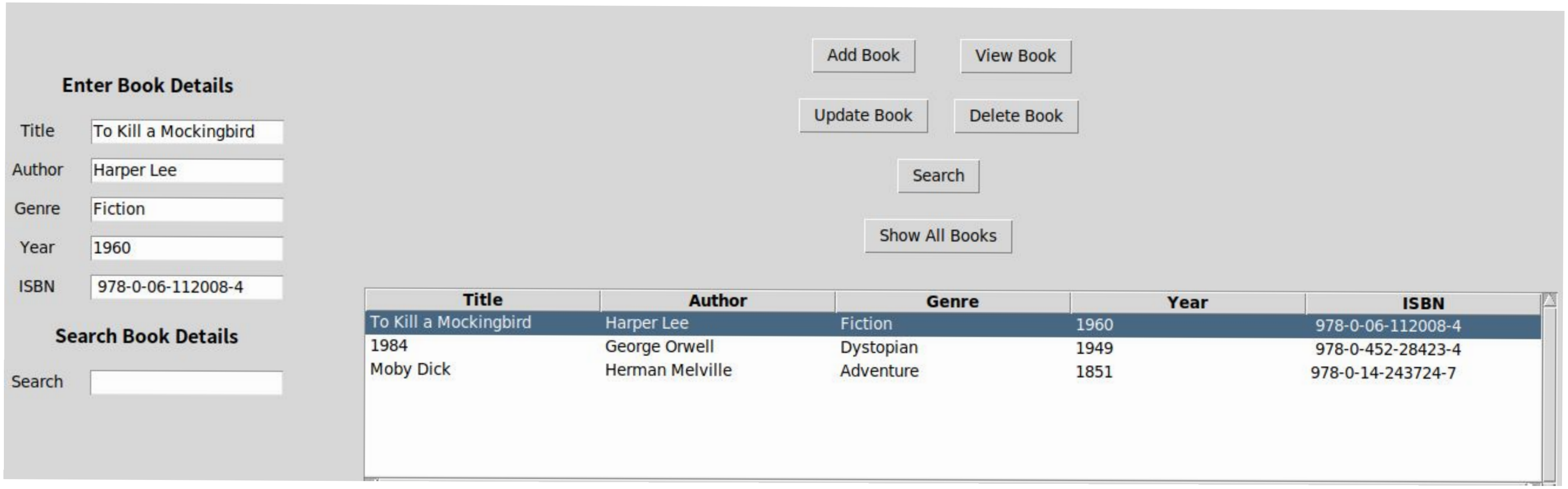
View Book

- The view_book function is designed to display the details of a selected book from the Treeview widget into the input fields of the GUI.
- **1.Selecting the Book in the Treeview**
- The focus() method returns the identifier (ID) of the currently selected item (row) in the Treeview widget.
- **2. Retrieving the Selected Item**
- Retrieves the currently selected item (row) in the Treeview widget.
- Gets the values (data) of the selected item.
- **3. Clearing Existing Entries**
- **4. Displaying Selected Book Details**
- entry_field.insert(0, selected_book[0]): Inserts the field of the selected book into the entry_namefield.
- These lines update the input fields with the details of the selected book.
- **5.Handling Errors**
- except IndexError:: Catches an IndexError if no item is selected or if there's an issue accessing the selected item



Please select a book to view its details.

Harper Lee	Fiction
George Orwell	Dystopian
Herman Melville	Adventure



Enter Book Details

Title	To Kill a Mockingbird
Author	Harper Lee
Genre	Fiction
Year	1960
ISBN	978-0-06-112008-4

Search Book Details

Search

Title	Author	Genre	Year	ISBN
To Kill a Mockingbird	Harper Lee	Fiction	1960	978-0-06-112008-4
1984	George Orwell	Dystopian	1949	978-0-452-28423-4
Moby Dick	Herman Melville	Adventure	1851	978-0-14-243724-7

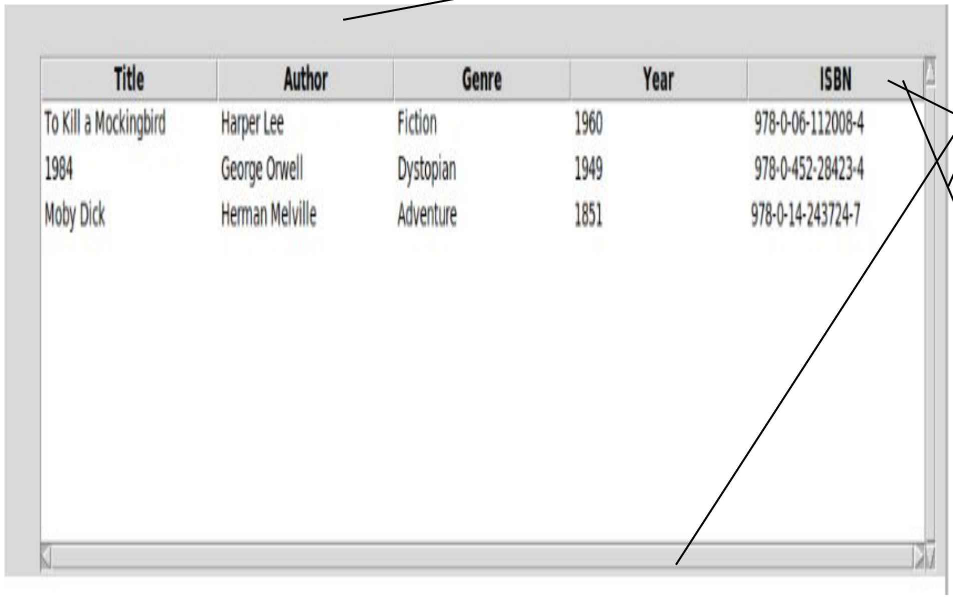
DELETE BOOK

- **Selecting the Book to Delete**
- retrieves the ID of the currently selected item
- returns a tuple of values corresponding to the selected item
- extracts the ISBN value from the tuple.
- **Executing the Delete Operation**
- The SQL DELETE statement removes the record that matches

SHOW ALL

- Clear the Treeview
- Query the Database for All Books
- 'SELECT * FROM Books'
- Display All Books in the Treeview
- If rows is empty, indicating that no books are found in the database, a message box (mb.showinfo()) notifies the user that there are no books to display.

Treeview for Displaying Books



Title	Author	Genre	Year	ISBN
To Kill a Mockingbird	Harper Lee	Fiction	1960	978-0-06-112008-4
1984	George Orwell	Dystopian	1949	978-0-452-28423-4
Moby Dick	Herman Melville	Adventure	1851	978-0-14-243724-7

- **tree_frame**: A frame is created to hold the Treeview widget and its scrollbars.
- **Vertical and horizontal scrollbars** are created and packed inside the frame
- **tree =**

Display Books Function and Main Event Loop

- display_books(): A function call to populate the Treeview with book data from the database when the application starts.
- **<u>EVENT LOOP</u>**
- **root.mainloop():** Starts the Tkinter event loop, waiting for user interactions.
- **<u>Database Connection Closure</u>**
- **connector.close():** Closes the database connection when the application exits.

CONCLUSION

- **Database Management**
- **User Interface Design**
- **Data Integrity and Validation**
- **Comprehensive Functionality**
- **Error Handling and User Feedback**

THANK YOU