

**A FIELD PROJECT REPORT ON  
BOOKSPIRE**

**Submitted**

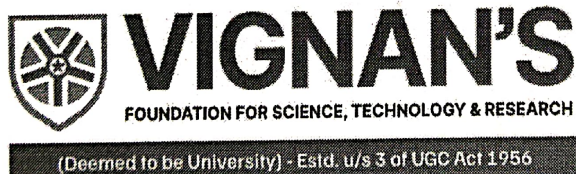
*In partial fulfilment of the requirements for the award of the degree*

**BACHELOR OF TECHNOLOGY  
in  
COMPUTER SCIENCE ENGINEERING**

By

S.H.V.S.ADITYA SEKHAR	231FA04934
J.DEVENDRA	231FA04966
L.HASIKKHA	231FA04A15
B.HARSHITHA	231FA04G95

Under the Guidance of  
**MR.T.Narasimha Rao**  
Assistant Professor, CSE



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING  
SCHOOL OF COMPUTING AND INFORMATICS  
VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH  
(Deemed to be University)  
Vadlamudi, Guntur, Andhra Pradesh-522213, India**

**April, 2025**

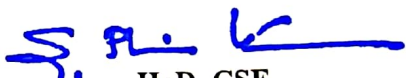
**CERTIFICATE**

This is to certify that the field project entitled "*BOOK SPIRE*" is being submitted by S.H.V.S.Aditya Sekhar, 231FA04934, J.Devendra, 231FA04966, L.Hasikkha, 231FA04A15, and B.Harshitha, 231FA04G95 in partial fulfilment of the requirements for the degree of Bachelor of Technology (B.Tech.) in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

This is a bonafide work carried out by the aforementioned students under my guidance and supervision.

  
Guide

  
Project Review Committee

  
HoD, CSE

HoD  
Dept. of Computer Science & Engineering  
VFSTR Deemed to be University  
VADLAMUDI - 522 213  
Guntur Dist., A.P., India.

**DECLARATION**

Date: 26/4/25

We hereby declare that our project work described in the field project titled "Book Spire" is the result of our own efforts and investigations

This project is being submitted under the supervision of **MR.T.Narasimha Rao**, Assistant Professor, Department of Computer Science and Engineering in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

S.H.V.S.ADITYA SEKHAR	231FA04934	Signature: S.H.V.S. Aditya Sekhar
J.DEVENDRA	231FA04966	Signature: J. Devendra
L.HASIKKHA	231FA04A15	Signature: L. Hasikkha
B.HARSHITHA	231FA04G95	Signature: B. Harshitha

## Contents

<b>Chapter No.</b>	<b>Description</b>	<b>Page No.</b>
<b>1</b> 1.1 1.2 1.3 1.4	<b>Introduction</b> <b>Problem Definition</b> <b>Existing System</b> <b>Proposed System</b> <b>Literature Review</b>	<b>5-6</b>
<b>2</b> 2.1 2.2	<b>System Requirements</b> <b>Hardware Requirements</b> <b>Software Requirements</b>	<b>7-8</b>
<b>3</b> 3.1 3.2	<b>System Design</b> <b>Modules of System</b> <b>UML Diagrams</b>	<b>9-12</b>
<b>4</b> 4.1 4.2	<b>Implementation</b> <b>Sample Code</b> <b>Test Cases</b>	<b>13-85</b>
<b>5</b> 5.1	<b>Results</b> <b>Output Screens</b>	<b>86-88</b>
<b>6</b>	<b>Conclusion</b>	<b>89</b>
<b>7</b>	<b>References</b>	<b>90</b>

# **1.Introduction**

BookSpire is a web-based platform designed for book lovers to explore, manage, and personalize their reading journey. It allows users to create profiles, maintain a reading list, and track their progress seamlessly. With a user-friendly interface, BookSpire ensures an engaging experience, making it easier to discover new books and stay organized. Whether you're an avid reader or just starting, BookSpire helps you keep your literary adventures in one place.

## **1.1.Problem Definition: -**

With the growing popularity of digital reading, book lovers need an efficient way to organize and track their reading progress. Traditional methods, such as maintaining handwritten lists or using spreadsheets, are time-consuming and inefficient. Existing book-tracking platforms often lack real-time updates, personalized recommendations, and a user-friendly interface, making it difficult for readers to manage their books seamlessly. Furthermore, users often struggle to keep track of books they have read, are currently reading, or plan to read in the future.

This project aims to develop BookSpire, an interactive web-based book management system that allows users to create and maintain their reading lists effortlessly. The system provides real-time updates, personalized book recommendations, and a dynamic user interface to enhance the reading experience. With secure user authentication, automated progress tracking, and an intuitive design, BookSpire ensures that users can efficiently manage their literary journey. The platform will ultimately improve book organization, encourage reading habits, and make book tracking an engaging and enjoyable experience.

## **1.2.Existing System: -**

Currently, book lovers rely on various platforms and manual methods to manage their reading lists, but these approaches come with several limitations. Many users maintain handwritten lists, spreadsheets, or notes to track their books, which can be inefficient and difficult to manage. Some existing book management platforms offer tracking features but lack real-time updates, personalized recommendations, and user-friendly interfaces.

**The existing system has the following key limitations:**

- Manual Tracking – Users must manually update their reading lists, making it time-consuming.
- Lack of Personalization – Existing platforms provide generic book recommendations that do not consider individual reading preferences.
- Limited Real-time Updates – Changes made to book status (e.g., “Currently Reading,” “Completed”) do not update instantly.
- Complex User Interfaces – Some platforms are cluttered and difficult to navigate, leading to a frustrating user experience.

Due to these challenges, there is a need for an improved book management system that provides an interactive, real-time, and personalized approach to tracking books and enhancing the reading experience.

### **1.3.Proposed System: -**

To address the limitations of traditional book-tracking methods, BookSpire introduces a more efficient, dynamic, and user-centric approach to book management. It enhances the user experience by automating updates, improving accessibility, and offering personalized features.

#### **Key Features of BookSpire:**

**Book Tracking** – Users can instantly add, update, or remove books from their reading lists.

**Dynamic Profile Updates** – The system updates user profiles dynamically, ensuring the latest information is displayed without manual refresh.

**Personalized Reading Lists** – Users can customize their reading lists based on preferences.

**Secure Authentication & Profile Management** – A secure login and signup system maintains user data.

**User-Friendly Interface** – Simplified UI/UX design ensures ease of navigation.

**Quick Access to Book Details** – Users can easily view book information, recommendations, and progress tracking.

### **1.4.Literature Review: -**

As we developed this project for the bookworms as to support their reading

Enthusiastic and to keep a place to track them. This project aim is to make the existing to be more performative and make the new ones to get attracted to the books and expressing their thoughts through reviews.

In developing this project we was developing a user friendly website which was having login interface such that their data can be seen for them. Through their data we focused on the TBR(“TO BE READ LIST”). This is the essential thing to keep them tracking

The Search interface makes the easier to the user to check the books by using genres that makes the user to discover the new books in the respective genre.

Overall, the project has been a thrilling experience that has allowed us to develop our technical skills and ability to work in teams. We have gained vast knowledge in web development, project planning, and effective solving of technical problems.

## **2.SYSTEM REQUIRMENTS**

### **2.1. HARDWARE REQUIRMENTS FOR BOOKSPIRE:**

To develop a BookSpire project, you will need to consider both hardware and software system requirements. Below is a general guide to the system requirements you may need.

<b>COMPONENT</b>	<b>MINIMUM REQUIRMENT</b>	<b>RECOMMENDED REQUIRMENT</b>
<b>Processor</b>	Intel i3 processor	Intel i5 or higher
<b>RAM</b>	4GB	4GB or more
<b>Storage</b>	350MB of free disk space	1GB or more
<b>Display</b>	1366 x768	1920 x 1080
<b>Keyboard &amp;Mouse</b>	May be required for website use	Required for website use
<b>Internet Connection</b>	Required for login and accessing wishlist of users,the available books	Required for login and accessing wishlist of users,the available books

## 2.2 SOFTWARE REQUIRMENTS FOR BOOKSPIRE:

COMPONENT	DETAILS
FRONTEND TECHNOLOGY	<p><b>HTML5:-</b> Structures the webpage and implements the book management interface.</p> <p><b>CSS3:-</b> Enhances the visual design and responsiveness of the interface.</p> <p><b>React (JSX):-</b> Provides a component-based UI for enhanced user experience.</p> <p><b>Vite :-</b>Modern build tool for fast development and optimized production.</p>
BACKEND TECHNOLOGY	<p><b>Database(MongoDB Atlas):</b> Serves as the database solution to manage user accounts and store book-related data securely.</p> <p><b>JavaScript:-</b> handles backend communication, including sending login and signup data to the server, processing responses, storing user session details using Redux, and navigating between pages based on authentication status.</p>
BROWSER COMPATABILITY	<ol style="list-style-type: none"><li>1. Google Chrome</li><li>2. Mozilla Firefox</li><li>3. Microsoft Edge</li><li>4. Safari</li></ol>
Text Editor	<p><b>Recommended:-</b></p> <p>VS Code, Sublime Text</p>



### **3.System Design**

#### **3.1.1. User Interface (UI) Module**

##### **Components:**

- layout.jsx → Defines the overall structure and layout of the application.
- app-sidebar.jsx → Provides navigation and menu options.
- BookCard.jsx → Displays individual book details in a card format.
- CreateBookForm.jsx → Allows users to add new books.

##### **Functionality:**

- Provides a visually appealing and structured user interface.
- Enhances navigation and user interaction with a dynamic sidebar.

#### **3.1.2. Pages Module**

##### **Components:**

- Auth.jsx → Manages user authentication (Login/Signup).
- Books.jsx → Displays a list of books available for the user.
- Wishlist.jsx → Allows users to view and manage their wishlist.

##### **Functionality:**

- Ensures smooth navigation between different sections of the application.
- Handles user authentication and book browsing.

#### **3.1.3. State Management Module**

##### **Files:**

- store.js → Configures Redux store for global state management.
- authSlice.js → Manages authentication state (login/logout functionality).

##### **Functionality:**

- Maintains a global state for authentication and user data.
- Provides centralized state management for consistent user experience.

### **3.1.4. Hooks Module**

#### **Files:**

- use-mobile.js → Custom hook for handling mobile responsiveness.

#### **Functionality:**

- Detects mobile viewports and adjusts UI accordingly.

### **3.1.5. Styling Module**

#### **Files:**

- App.css → Global styles for the application.
- index.css → Additional styles for layout and elements.

#### **Functionality:**

- Ensures a responsive and visually appealing design.

### **3.1.6. Entry Point & Configuration Module**

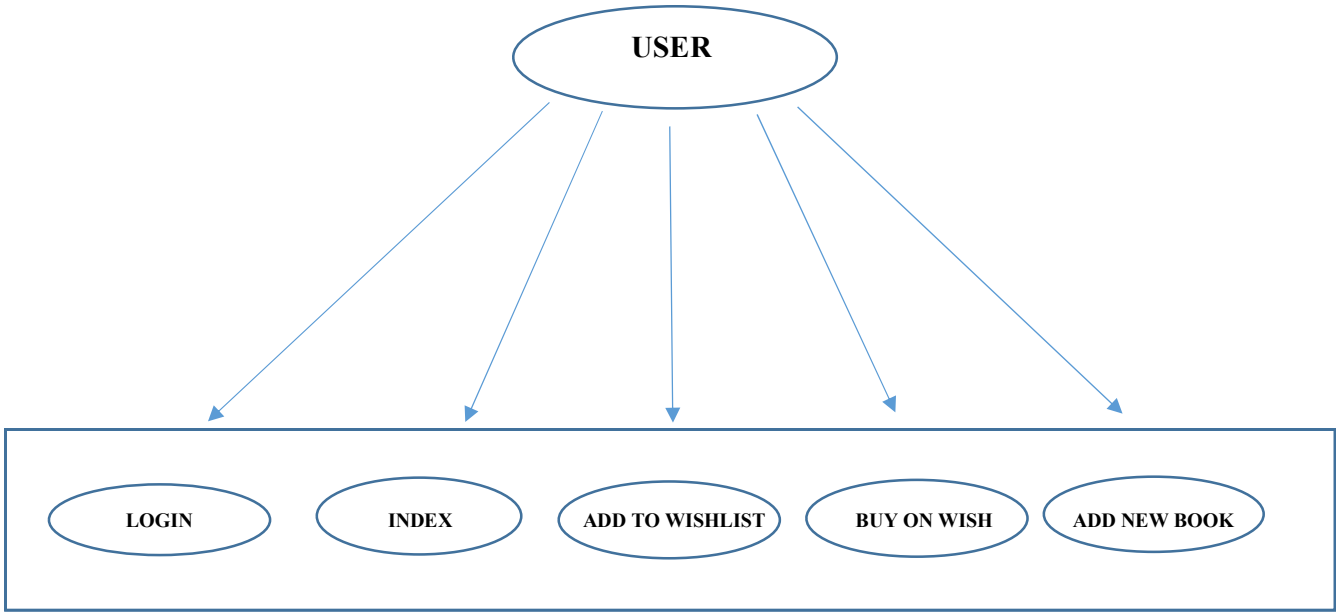
#### **Files:**

- main.jsx → Renders the root React component.
- App.jsx → Defines the main application structure.
- index.html → The base HTML file where the React app is injected.
- .gitignore → Excludes unnecessary files from version control.
- package.json → Manages project dependencies.
- eslint.config.js → Enforces coding standards.

#### **Functionality:**

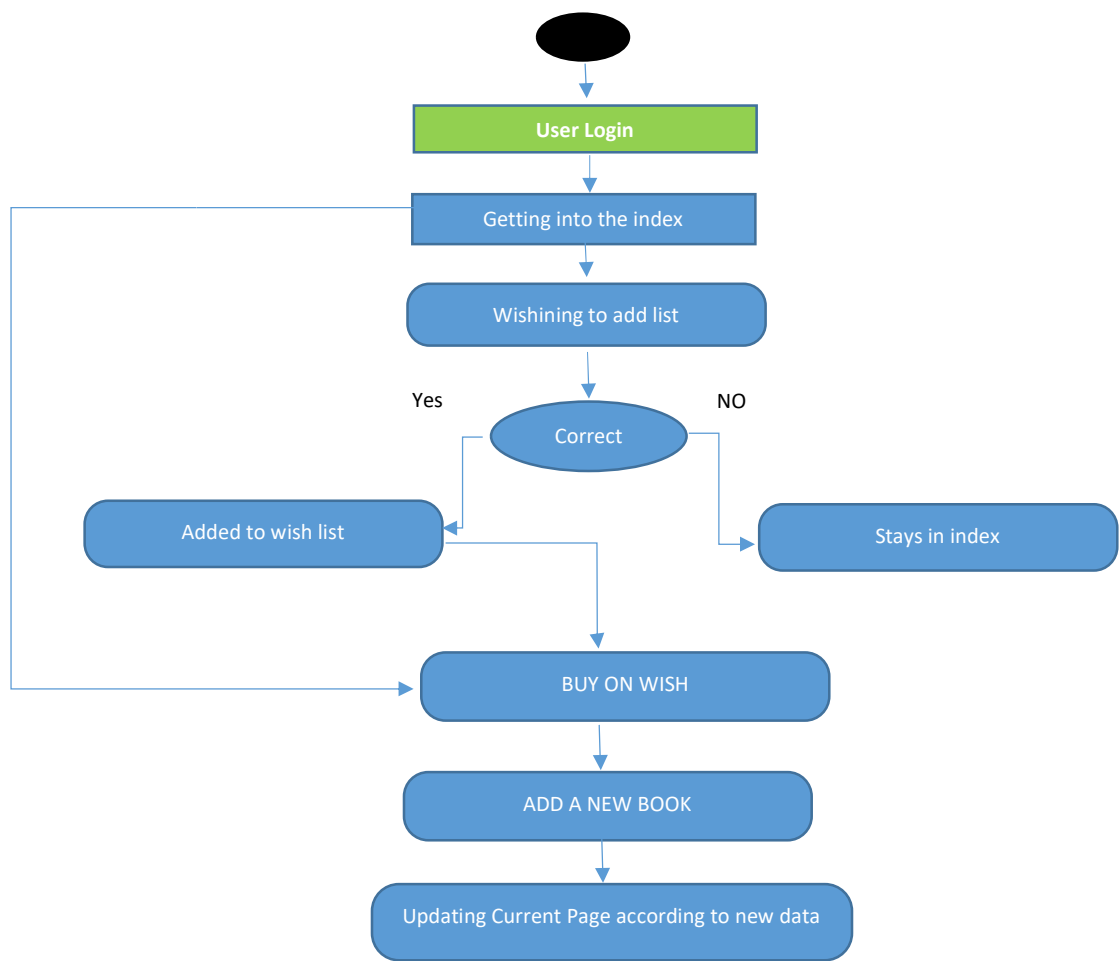
- Initializes the application and ensures proper configuration.
- Provides dependency management and code quality enforcement.

**3.2: UML DIAGRAMS:**



**Fig: -3.2.1**

**Use Case Diagram**



**Fig: -3.2.2**

## Activity Diagram

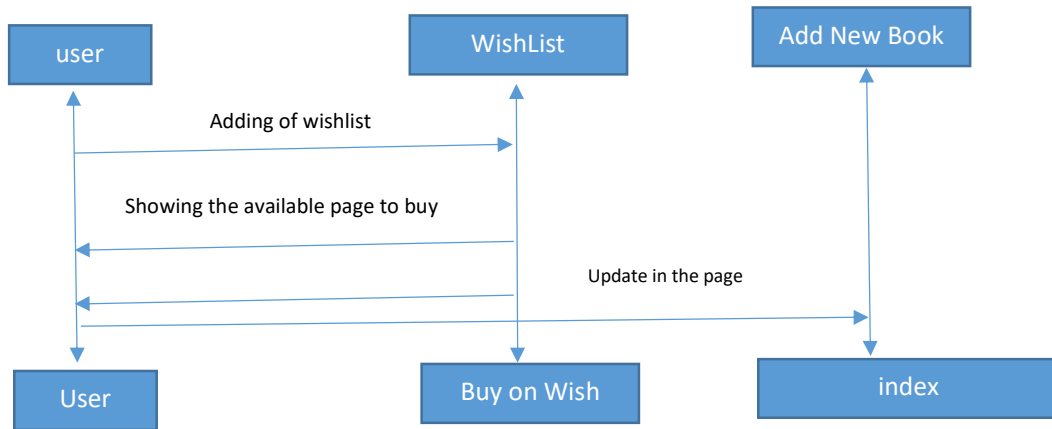


Fig: -3.2.3

## Sequence Diagram

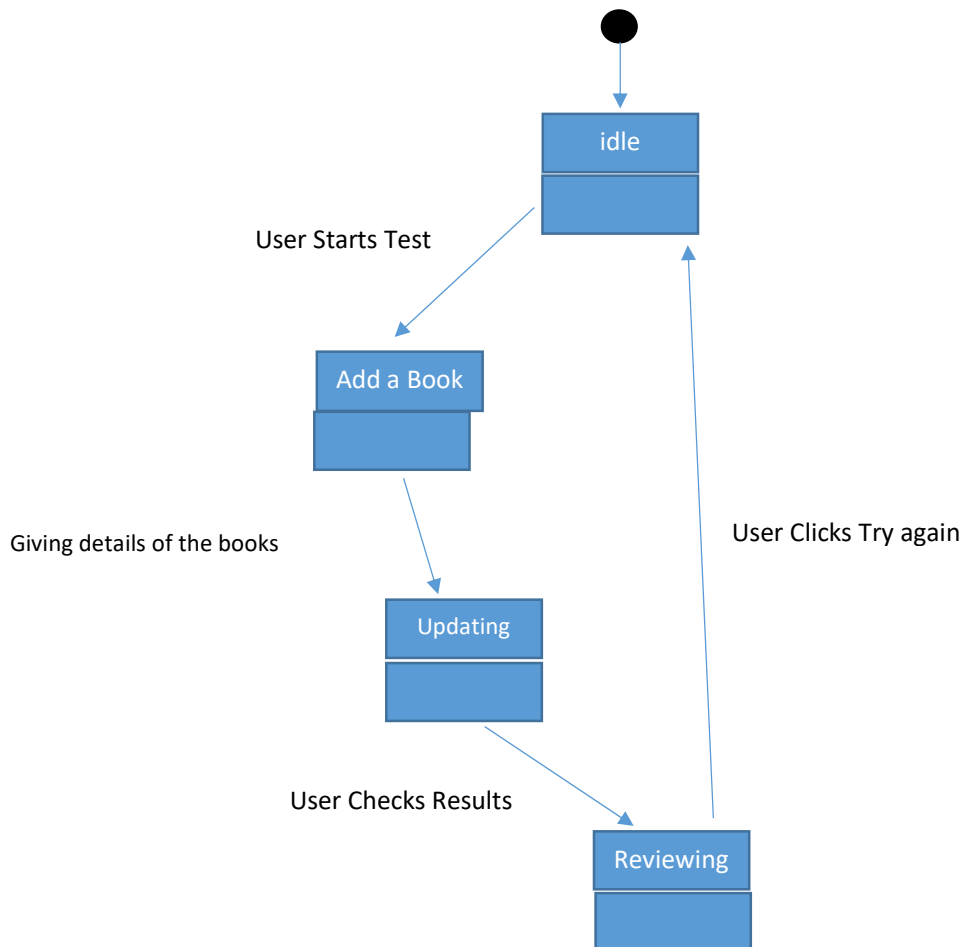


Fig: -3.2.4

## 4.Implementation

### 4.1) Code

#### Frontend:-

##### Index.html:-

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>BookSpire-Home</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

#### Ui: -

##### src\componets\ui\Button.jsx:-

```
import * as React from "react"
import { Slot } from "@radix-ui/react-slot"
import { cva } from "class-variance-authority";
import { cn } from "@lib/utlis"
const buttonVariants = cva(
  "inline-flex items-center justify-center gap-2 whitespace-nowrap rounded-md text-sm font-medium transition-all disabled:pointer-events-none disabled:opacity-50 [&_svg]:pointer-events-none [&_svg:not([class*='size-'])]:size-4 shrink-0 [&_svg]:shrink-0 outline-none focus-visible:border-ring focus-visible:ring-ring/50 focus-visible:ring-[3px] aria-invalid:ring-destructive/20 dark:aria-invalid:ring-destructive/40 aria-invalid:border-destructive",
  {
    variants: {
```

```

variant: {
  default:
    "bg-primary text-primary-foreground shadow-xs hover:bg-primary/90",
  destructive:
    "bg-destructive text-white shadow-xs hover:bg-destructive/90 focus-visible:ring-destructive/20 dark:focus-visible:ring-destructive/40 dark:bg-destructive/60",
  outline:
    "border bg-background shadow-xs hover:bg-accent hover:text-accent-foreground dark:bg-input/30 dark:border-input dark:hover:bg-input/50",
  secondary:
    "bg-secondary text-secondary-foreground shadow-xs hover:bg-secondary/80",
  ghost:
    "hover:bg-accent hover:text-accent-foreground dark:hover:bg-accent/50",
  link: "text-primary underline-offset-4 hover:underline",
},
size: {
  default: "h-9 px-4 py-2 has- [>svg]:px-3",
  sm: "h-8 rounded-md gap-1.5 px-3 has- [>svg]:px-2.5",
  lg: "h-10 rounded-md px-6 has- [>svg]:px-4",
  icon: "size-9",
},
},
defaultVariants: {
  variant: "default",
  size: "default",
},
}
)

function Button({
  className,
  variant,
  size,

```

```

    asChild = false,
    ...props
  }) {
    const Comp = asChild ? Slot : "button"

    return (
      <Comp
        data-slot="button"
        className={cn(buttonVariants({ variant, size, className })))}
        {...props} />
    );
  }
export { Button, buttonVariants }

```

**src\components\ui\card.jsx:-**

```

import * as React from "react"
import { cn } from "@lib/utls"

function Card({
  className,
  ...props
}) {
  return (
    <div
      data-slot="card"
      className={cn(
        "bg-card text-card-foreground flex flex-col gap-6 rounded-xl border py-6 shadow-sm",
        className
      )}
      {...props} />
  );
}

function CardHeader({

```

```

    className,
    ...props
  }) {
    return (
      <div
        data-slot="card-header"
        className={cn(
          "@container/card-header grid auto-rows-min grid-rows-[auto_auto] items-start gap-1.5
px-6 has-data-[slot=card-action]:grid-cols-[1fr_auto] [.border-b]:pb-6",
          className
        )}
        {...props} />
      );
    }
  }

function CardTitle({
  className,
  ...props
}) {
  return (
    <div
      data-slot="card-title"
      className={cn("leading-none font-semibold", className)}
      {...props} />
    );
  }
}

function CardDescription({
  className,
  ...props
}) {
  return (
    <div

```



```

      data-slot="card-description"
      className={cn("text-muted-foreground text-sm", className)}
      {...props} />)
    );
  }
function CardAction({
  className,
  ...props
}) {
  return (
    <div
      data-slot="card-action"
      className={cn(
        "col-start-2 row-span-2 row-start-1 self-start justify-self-end",
        className
      )}
      {...props} />)
  );
}
function CardContent({
  className,
  ...props
}) {
  return (<div data-slot="card-content" className={cn("px-6", className)} {...props} />);
}
function CardFooter({
  className,
  ...props
}) {
  return (
    <div

```

```

    data-slot="card-footer"

    className={cn("flex items-center px-6 [.border-t]:pt-6", className)}

    {...props} />)
  );
}

export {
  Card,
  CardHeader,
  CardFooter,
  CardTitle,
  CardAction,
  CardDescription,
  CardContent,
}

```

#### **src\components\ui\Input.jsx:-**

```

import * as React from "react"
import { cn } from "@lib/utls"

function Input({
  className,
  type,
  ...props
}) {
  return (
    <input
      type={type}
      data-slot="input"
      className={cn(
        "file:text-foreground placeholder:text-muted-foreground selection:bg-primary
        selection:text-primary-foreground dark:bg-input/30 border-input flex h-9 w-full min-w-0
        rounded-md border bg-transparent px-3 py-1 text-base shadow-xs transition-[color,box-
        shadow] outline-none file:inline-flex file:h-7 file:border-0 file:bg-transparent file:text-sm
        file:font-medium disabled:pointer-events-none disabled:cursor-not-allowed disabled:opacity-
        50 md:text-sm",

```

```

    "focus-visible:border-ring focus-visible:ring-ring/50 focus-visible:ring-[3px]",
    "aria-invalid:ring-destructive/20 dark:aria-invalid:ring-destructive/40 aria-
invalid:border-destructive",
    className
  )}
  {...props} />)
);
}
export { Input }

```

src/components/ui/separator.jsx:-

```

"use client"

import * as React from "react"
import * as SeparatorPrimitive from "@radix-ui/react-separator"
import { cn } from "@lib/utls"

function Separator({
  className,
  orientation = "horizontal",
  decorative = true,
  ...props
}) {
  return (
    (<SeparatorPrimitive.Root
      data-slot="separator-root"
      decorative={decorative}
      orientation={orientation}
      className={cn(
        "bg-border shrink-0 data-[orientation=horizontal]:h-px data-[orientation=horizontal]:w-
full data-[orientation=vertical]:h-full data-[orientation=vertical]:w-px",
        className
      )}
    {...props} />)
  )
}

```

```

    );
  }
  export { Separator }
components\ui\sheet.jsx:-
import * as React from "react"
import * as SheetPrimitive from "@radix-ui/react-dialog"
import { XIcon } from "lucide-react"

import { cn } from "@lib/utls"

function Sheet({
  ...props
}) {
  return <SheetPrimitive.Root data-slot="sheet" {...props} />;
}

function SheetTrigger({
  ...props
}) {
  return <SheetPrimitive.Trigger data-slot="sheet-trigger" {...props} />;
}

function SheetClose({
  ...props
}) {
  return <SheetPrimitive.Close data-slot="sheet-close" {...props} />;
}

function SheetPortal({
  ...props
}) {

```

```

    return <SheetPrimitive.Portal data-slot="sheet-portal" {...props} />;
  }

function SheetOverlay({
  className,
  ...props
}) {
  return (
    <SheetPrimitive.Overlay
      data-slot="sheet-overlay"
      className={cn(
        "data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 fixed inset-0 z-50 bg-black/50",
        className
      )}
      {...props} />)
  );
}

```

```

function SheetContent({
  className,
  children,
  side = "right",
  ...props
}) {
  return (
    <SheetPortal>
      <SheetOverlay />
      <SheetPrimitive.Content
        data-slot="sheet-content"
        className={cn(

```

```
"bg-background data-[state=open]:animate-in data-[state=closed]:animate-out fixed z-50 flex flex-col gap-4 shadow-lg transition ease-in-out data-[state=closed]:duration-300 data-[state=open]:duration-500",
```

```
side === "right" &&
```

```
"data-[state=closed]:slide-out-to-right data-[state=open]:slide-in-from-right inset-y-0 right-0 h-full w-3/4 border-l sm:max-w-sm",
```

```
side === "left" &&
```

```
"data-[state=closed]:slide-out-to-left data-[state=open]:slide-in-from-left inset-y-0 left-0 h-full w-3/4 border-r sm:max-w-sm",
```

```
side === "top" &&
```

```
"data-[state=closed]:slide-out-to-top data-[state=open]:slide-in-from-top inset-x-0 top-0 h-auto border-b",
```

```
side === "bottom" &&
```

```
"data-[state=closed]:slide-out-to-bottom data-[state=open]:slide-in-from-bottom inset-x-0 bottom-0 h-auto border-t",
```

```
className
```

```
))
```

```
{...props}>
```

```
{children}
```

```
<SheetPrimitive.Close
```

```
className="ring-offset-background focus:ring-ring data-[state=open]:bg-secondary absolute top-4 right-4 rounded-xs opacity-70 transition-opacity hover:opacity-100 focus:ring-2 focus:ring-offset-2 focus:outline-hidden disabled:pointer-events-none">
```

```
<XIcon className="size-4" />
```

```
<span className="sr-only">Close</span>
```

```
</SheetPrimitive.Close>
```

```
</SheetPrimitive.Content>
```

```
</SheetPortal>)
```

```
);
```

```
}
```

```
function SheetHeader({
```

```
  className,
```

```
  ...props
```

```

    }) {
      return (
        <div
          data-slot="sheet-header"
          className={cn("flex flex-col gap-1.5 p-4", className)}
          {...props} />
        );
      }

```

```

function SheetFooter({
  className,
  ...props
}) {
  return (
    <div
      data-slot="sheet-footer"
      className={cn("mt-auto flex flex-col gap-2 p-4", className)}
      {...props} />
    );
  }

```

```

function SheetTitle({
  className,
  ...props
}) {
  return (
    <SheetPrimitive.Title
      data-slot="sheet-title"
      className={cn("text-foreground font-semibold", className)}
      {...props} />
    );
  }

```

```
}
```

```
function SheetDescription({
  className,
  ...props
}) {
  return (
    (<SheetPrimitive.Description
      data-slot="sheet-description"
      className={cn("text-muted-foreground text-sm", className)}
      {...props} />)
  );
}
```

```
export {
  Sheet,
  SheetTrigger,
  SheetClose,
  SheetContent,
  SheetHeader,
  SheetFooter,
  SheetTitle,
  SheetDescription,
}
```

#### **components\ui\Sidebar.jsx:-**

```
import * as React from "react"
import { Slot } from "@radix-ui/react-slot"
import { cva } from "class-variance-authority";
import { PanelLeftIcon } from "lucide-react"

import { useIsMobile } from "@hooks/use-mobile"
```



```

import { cn } from "@lib/utils"

import { Button } from "@components/ui/button"
import { Input } from "@components/ui/input"
import { Separator } from "@components/ui/separator"
import {
  Sheet,
  SheetContent,
  SheetDescription,
  SheetHeader,
  SheetTitle,
} from "@components/ui/sheet"
import { Skeleton } from "@components/ui/skeleton"
import {
  Tooltip,
  TooltipContent,
  TooltipProvider,
  TooltipTrigger,
} from "@components/ui/tooltip"

```

```

const SIDEBAR_COOKIE_NAME = "sidebar_state"
const SIDEBAR_COOKIE_MAX_AGE = 60 * 60 * 24 * 7
const SIDEBAR_WIDTH = "16rem"
const SIDEBAR_WIDTH_MOBILE = "18rem"
const SIDEBAR_WIDTH_ICON = "3rem"
const SIDEBAR_KEYBOARD_SHORTCUT = "b"

```

```

const SidebarContext = React.createContext(null)

```

```

function useSidebar() {
  const context = React.useContext(SidebarContext)
  if (!context) {

```

```

    throw new Error("useSidebar must be used within a SidebarProvider.")
  }
  return context
}

function SidebarProvider({
  defaultOpen = true,
  open: openProp,
  onOpenChange: setOpenProp,
  className,
  style,
  children,
  ...props
}) {
  const isMobile = useIsMobile()
  const [openMobile, setOpenMobile] = React.useState(false)

  // This is the internal state of the sidebar.
  // We use openProp and setOpenProp for control from outside the component.
  const [_open, _setOpen] = React.useState(defaultOpen)
  const open = openProp ?? _open
  const setOpen = React.useCallback((value) => {
    const openState = typeof value === "function" ? value(open) : value
    if (setOpenProp) {
      setOpenProp(openState)
    } else {
      _setOpen(openState)
    }
  })

  // This sets the cookie to keep the sidebar state.
  document.cookie = `${SIDEBAR_COOKIE_NAME}=${openState}; path=/; max-age=${SIDEBAR_COOKIE_MAX_AGE}`

```

```

}, [setOpenProp, open])

// Helper to toggle the sidebar.
const toggleSidebar = React.useCallback(() => {
  return isMobile ? setOpenMobile((open) => !open) : setOpen((open) => !open);
}, [isMobile, setOpen, setOpenMobile])

// Adds a keyboard shortcut to toggle the sidebar.
React.useEffect(() => {
  const handleKeyDown = (event) => {
    if (
      event.key === SIDEBAR_KEYBOARD_SHORTCUT &&
      (event.metaKey || event.ctrlKey)
    ) {
      event.preventDefault()
      toggleSidebar()
    }
  }

  window.addEventListener("keydown", handleKeyDown)

  return () => window.removeEventListener("keydown", handleKeyDown);
}, [toggleSidebar])

// We add a state so that we can do data-state="expanded" or "collapsed".
// This makes it easier to style the sidebar with Tailwind classes.
const state = open ? "expanded" : "collapsed"

const contextValue = React.useMemo(() => ({
  state,
  open,
  setOpen,

```

```

isMobile,
openMobile,
setOpenMobile,
toggleSidebar,
}), [state, open, setOpen, isMobile, openMobile, setOpenMobile, toggleSidebar])

```

```

return (
  (<SidebarContext.Provider value={contextValue}>
    <TooltipProvider delayDuration={0}>
      <div
        data-slot="sidebar-wrapper"
        style={
          {
            "--sidebar-width": SIDEBAR_WIDTH,
            "--sidebar-width-icon": SIDEBAR_WIDTH_ICON,
            ...style
          }
        }
        className={cn(
          "group/sidebar-wrapper has-data-[variant=inset]:bg-sidebar flex min-h-svh w-full",
          className
        )}
        {...props}>
        {children}
      </div>
    </TooltipProvider>
  </SidebarContext.Provider>)
);
}

```

```

function Sidebar({

```

```

side = "left",
variant = "sidebar",
collapsible = "offcanvas",
className,
children,
...props
})) {
  const { isMobile, state, openMobile, setOpenMobile } = useSidebar()

  if (collapsible === "none") {
    return (
      <div
        data-slot="sidebar"
        className={cn(
          "bg-sidebar text-sidebar-foreground flex h-full w-(--sidebar-width) flex-col",
          className
        )}
        {...props}>
        {children}
      </div>
    );
  }

  if (isMobile) {
    return (
      <Sheet open={openMobile} onOpenChange={setOpenMobile} {...props}>
        <SheetContent
          data-sidebar="sidebar"
          data-slot="sidebar"
          data-mobile="true"

```

```

        className="bg-sidebar text-sidebar-foreground w-(--sidebar-width) p-0
[&>button]:hidden"
        style={
            {
                "--sidebar-width": SIDEBAR_WIDTH_MOBILE
            }
        }
        side={side}>
        <SheetHeader className="sr-only">
            <SheetTitle>Sidebar</SheetTitle>
            <SheetDescription>Displays the mobile sidebar.</SheetDescription>
        </SheetHeader>
        <div className="flex h-full w-full flex-col">{children}</div>
    </SheetContent>
</Sheet>)
);
}

return (
    <div
        className="group peer text-sidebar-foreground hidden md:block"
        data-state={state}
        data-collapsible={state === "collapsed" ? collapsible : ""}
        data-variant={variant}
        data-side={side}
        data-slot="sidebar">
        {/* This is what handles the sidebar gap on desktop */}
        <div
            data-slot="sidebar-gap"
            className={cn(
                "relative w-(--sidebar-width) bg-transparent transition-[width] duration-200 ease-
linear",

```

```

    "group-data-[collapsible=offcanvas]:w-0",
    "group-data-[side=right]:rotate-180",
    variant === "floating" || variant === "inset"
      ? "group-data-[collapsible=icon]:w-[calc(var(--sidebar-width-icon)+(--spacing(4)))]"
      : "group-data-[collapsible=icon]:w-(--sidebar-width-icon)"
  )} />
<div
  data-slot="sidebar-container"
  className={cn(
    "fixed inset-y-0 z-10 hidden h-svh w-(--sidebar-width) transition-[left,right,width]
    duration-200 ease-linear md:flex",
    side === "left"
      ? "left-0 group-data-[collapsible=offcanvas]:left-[calc(var(--sidebar-width)*-1)]"
      : "right-0 group-data-[collapsible=offcanvas]:right-[calc(var(--sidebar-width)*-1)]",
    // Adjust the padding for floating and inset variants.
    variant === "floating" || variant === "inset"
      ? "p-2 group-data-[collapsible=icon]:w-[calc(var(--sidebar width-icon)+(--
spacing(4))+2px)]"
      : "group-data-[collapsible=icon]:w-(--sidebar-width-icon) group-data-
[side=left]:border-r group-data-[side=right]:border-l",
    className
  )}
  {...props}>
<div
  data-sidebar="sidebar"
  data-slot="sidebar-inner"
  className="bg-sidebar group-data-[variant=floating]:border-sidebar-border flex h-full
w-full flex-col group-data-[variant=floating]:rounded-lg group-data-[variant=floating]:border
group-data-[variant=floating]:shadow-sm">
  {children}
</div>
</div>
</div>)

```

```
);  
}
```

```
function SidebarTrigger({  
  className,  
  onClick,  
  ...props  
}) {  
  const { toggleSidebar } = useSidebar()  
  
  return (  
    <Button  
      data-sidebar="trigger"  
      data-slot="sidebar-trigger"  
      variant="ghost"  
      size="icon"  
      className={cn("size-7", className)}  
      onClick={(event) => {  
        onClick?.(event)  
        toggleSidebar()  
      }}  
      {...props}>  
      <PanelLeftIcon />  
      <span className="sr-only">Toggle Sidebar</span>  
    </Button>  
  );  
}
```

```
function SidebarRail({  
  className,  
  ...props
```



```

    }) {
      const { toggleSidebar } = useSidebar()

      return (
        <button
          data-sidebar="rail"
          data-slot="sidebar-rail"
          aria-label="Toggle Sidebar"
          tabIndex={-1}
          onClick={toggleSidebar}
          title="Toggle Sidebar"
          className={cn(
            "hover:after:bg-sidebar-border absolute inset-y-0 z-20 hidden w-4 -translate-x-1/2
            transition-all ease-linear group-data-[side=left]:-right-4 group-data-[side=right]:left-0
            after:absolute after:inset-y-0 after:left-1/2 after:w-[2px] sm:flex",
            "in-data-[side=left]:cursor-w-resize in-data-[side=right]:cursor-e-resize",
            "[[data-side=left][data-state=collapsed]_&]:cursor-e-resize [[data-side=right][data-
            state=collapsed]_&]:cursor-w-resize",
            "hover:group-data-[collapsible=offcanvas]:bg-sidebar group-data-
            [collapsible=offcanvas]:translate-x-0 group-data-[collapsible=offcanvas]:after:left-full",
            "[[data-side=left][data-collapsible=offcanvas]_&]:-right-2",
            "[[data-side=right][data-collapsible=offcanvas]_&]:-left-2",
            className
          )}
        >
          {...props} />
        </button>
      )
    }
  }

  function SidebarInset({
    className,
    ...props
  }) {

```

```

return (
  (<main
    data-slot="sidebar-inset"
    className={cn(
      "bg-background relative flex w-full flex-1 flex-col",
      "md:peer-data-[variant=inset]:m-2 md:peer-data-[variant=inset]:ml-0 md:peer-data-[variant=inset]:rounded-xl md:peer-data-[variant=inset]:shadow-sm md:peer-data-[variant=inset]:peer-data-[state=collapsed]:ml-2",
      className
    )}
    {...props} />)
  );
}

```

```

function SidebarInput({
  className,
  ...props
}) {
  return (
    (<Input
      data-slot="sidebar-input"
      data-sidebar="input"
      className={cn("bg-background h-8 w-full shadow-none", className)}
      {...props} />)
    );
}

```

```

function SidebarHeader({
  className,
  ...props
}) {
  return (

```

```

(<div
  data-slot="sidebar-header"
  data-sidebar="header"
  className={cn("flex flex-col gap-2 p-2", className)}
  {...props} />)
);
}

```

```

function SidebarFooter({
  className,
  ...props
}) {
  return (
    <div
      data-slot="sidebar-footer"
      data-sidebar="footer"
      className={cn("flex flex-col gap-2 p-2", className)}
      {...props} />)
  );
}

```

```

function SidebarSeparator({
  className,
  ...props
}) {
  return (
    <Separator
      data-slot="sidebar-separator"
      data-sidebar="separator"
      className={cn("bg-sidebar-border mx-2 w-auto", className)}
      {...props} />)
  );
}

```

```
);  
}
```

```
function SidebarContent({  
  className,  
  ...props  
}) {  
  return (  
    <div  
      data-slot="sidebar-content"  
      data-sidebar="content"  
      className={cn(  
        "flex min-h-0 flex-1 flex-col gap-2 overflow-auto group-data-  
[collapsible=icon]:overflow-hidden",  
        className  
      )}  
      {...props} />  
    );  
  }  
}
```

```
function SidebarGroup({  
  className,  
  ...props  
}) {  
  return (  
    <div  
      data-slot="sidebar-group"  
      data-sidebar="group"  
      className={cn("relative flex w-full min-w-0 flex-col p-2", className)}  
      {...props} />  
    );  
  }  
}
```

```
}
```

```
function SidebarGroupLabel({
  className,
  asChild = false,
  ...props
}) {
  const Comp = asChild ? Slot : "div"

  return (
    (<Comp
      data-slot="sidebar-group-label"
      data-sidebar="group-label"
      className={cn(
        "text-sidebar-foreground/70 ring-sidebar-ring flex h-8 shrink-0 items-center rounded-md
        px-2 text-xs font-medium outline-hidden transition-[margin,opacity] duration-200 ease-linear
        focus-visible:ring-2 [&>svg]:size-4 [&>svg]:shrink-0",
        "group-data-[collapsible=icon]:-mt-8 group-data-[collapsible=icon]:opacity-0",
        className
      )}
      {...props} />)
  );
}
```

```
function SidebarGroupAction({
  className,
  asChild = false,
  ...props
}) {
  const Comp = asChild ? Slot : "button"

  return (
```

```

(<Comp
  data-slot="sidebar-group-action"
  data-sidebar="group-action"
  className={cn(
    "text-sidebar-foreground ring-sidebar-ring hover:bg-sidebar-accent hover:text-sidebar-
    accent-foreground absolute top-3.5 right-3 flex aspect-square w-5 items-center justify-center
    rounded-md p-0 outline-hidden transition-transform focus-visible:ring-2 [>svg]:size-4
    [>svg]:shrink-0",
    // Increases the hit area of the button on mobile.
    "after:absolute after:-inset-2 md:after:hidden",
    "group-data-[collapsible=icon]:hidden",
    className
  )}
  {...props} />)
);
}

```

```

function SidebarGroupContent({
  className,
  ...props
}) {
  return (
    <div
      data-slot="sidebar-group-content"
      data-sidebar="group-content"
      className={cn("w-full text-sm", className)}
      {...props} />
    );
}

```

```

function SidebarMenu({
  className,

```

```

    ...props
  }) {
    return (
      <ul
        data-slot="sidebar-menu"
        data-sidebar="menu"
        className={cn("flex w-full min-w-0 flex-col gap-1", className)}
        {...props} />
    );
  }

```

```

function SidebarMenuItem({
  className,
  ...props
}) {
  return (
    <li
      data-slot="sidebar-menu-item"
      data-sidebar="menu-item"
      className={cn("group/menu-item relative", className)}
      {...props} />
    );
  }

```

```
const sidebarMenuButtonVariants = cva(
```

```

  "peer/menu-button flex w-full items-center gap-2 overflow-hidden rounded-md p-2 text-left
  text-sm outline-hidden ring-sidebar-ring transition-[width,height,padding] hover:bg-sidebar-
  accent hover:text-sidebar-accent-foreground focus-visible:ring-2 active:bg-sidebar-accent
  active:text-sidebar-accent-foreground disabled:pointer-events-none disabled:opacity-50
  group-has-data-[sidebar=menu-action]/menu-item:pr-8 aria-disabled:pointer-events-none
  aria-disabled:opacity-50 data-[active=true]:bg-sidebar-accent data-[active=true]:font-medium
  data-[active=true]:text-sidebar-accent-foreground data-[state=open]:hover:bg-sidebar-accent
  data-[state=open]:hover:text-sidebar-accent-foreground group-data-[collapsible=icon]:size-8!
```

```

group-data-[collapsible=icon]:p-2! [&>span:last-child]:truncate [&>svg]:size-4
[&>svg]:shrink-0",
{
  variants: {
    variant: {
      default: "hover:bg-sidebar-accent hover:text-sidebar-accent-foreground",
      outline:
        "bg-background shadow-[0_0_0_1px_hsl(var(--sidebar-border))] hover:bg-sidebar-
        accent hover:text-sidebar-accent-foreground hover:shadow-[0_0_0_1px_hsl(var(--sidebar-
        accent))]",
    },
    size: {
      default: "h-8 text-sm",
      sm: "h-7 text-xs",
      lg: "h-12 text-sm group-data-[collapsible=icon]:p-0!",
    },
  },
  defaultVariants: {
    variant: "default",
    size: "default",
  },
}
)

```

```

function SidebarMenuButton({
  asChild = false,
  isActive = false,
  variant = "default",
  size = "default",
  tooltip,
  className,
  ...props

```



```

    }) {
      const Comp = asChild ? Slot : "button"
      const { isMobile, state } = useSidebar()

      const button = (
        <Comp
          data-slot="sidebar-menu-button"
          data-sidebar="menu-button"
          data-size={size}
          data-active={isActive}
          className={cn(sidebarMenuButtonVariants({ variant, size }), className)}
          {...props} />
        )

      if (!tooltip) {
        return button
      }

      if (typeof tooltip === "string") {
        tooltip = {
          children: tooltip,
        }
      }

      return (
        (<Tooltip>
          <TooltipTrigger asChild>{button}</TooltipTrigger>
          <TooltipContent
            side="right"
            align="center"
            hidden={state !== "collapsed" || isMobile}

```

```

    {...tooltip} />
  </Tooltip>)
);
}

```

```

function SidebarMenuAction({
  className,
  asChild = false,
  showOnHover = false,
  ...props
}) {
  const Comp = asChild ? Slot : "button"

  return (
    <Comp
      data-slot="sidebar-menu-action"
      data-sidebar="menu-action"
      className={cn(
        "text-sidebar-foreground ring-sidebar-ring hover:bg-sidebar-accent hover:text-sidebar-
        accent-foreground peer-hover/menu-button:text-sidebar-accent-foreground absolute top-1.5
        right-1 flex aspect-square w-5 items-center justify-center rounded-md p-0 outline-hidden
        transition-transform focus-visible:ring-2 [&>svg]:size-4 [&>svg]:shrink-0",
        // Increases the hit area of the button on mobile.
        "after:absolute after:-inset-2 md:after:hidden",
        "peer-data-[size=sm]/menu-button:top-1",
        "peer-data-[size=default]/menu-button:top-1.5",
        "peer-data-[size=lg]/menu-button:top-2.5",
        "group-data-[collapsible=icon]:hidden",
        showOnHover &&
        "peer-data-[active=true]/menu-button:text-sidebar-accent-foreground group-focus-
        within/menu-item:opacity-100 group-hover/menu-item:opacity-100 data-
        [state=open]:opacity-100 md:opacity-0",
        className

```

```

    })
    {...props} />)
  );
}

```

```

function SidebarMenuBadge({
  className,
  ...props
}) {
  return (
    <div
      data-slot="sidebar-menu-badge"
      data-sidebar="menu-badge"
      className={cn(
        "text-sidebar-foreground pointer-events-none absolute right-1 flex h-5 min-w-5 items-
center justify-center rounded-md px-1 text-xs font-medium tabular-nums select-none",
        "peer-hover/menu-button:text-sidebar-accent-foreground peer-data-[active=true]/menu-
button:text-sidebar-accent-foreground",
        "peer-data-[size=sm]/menu-button:top-1",
        "peer-data-[size=default]/menu-button:top-1.5",
        "peer-data-[size=lg]/menu-button:top-2.5",
        "group-data-[collapsible=icon]:hidden",
        className
      )}
      {...props} />)
  );
}

```

```

function SidebarMenuSkeleton({
  className,
  showIcon = false,
  ...props

```

```

    }) {
      // Random width between 50 to 90%.
      const width = React.useMemo(() => {
        return `${Math.floor(Math.random() * 40) + 50}%`;
      }, [])

      return (
        <div
          data-slot="sidebar-menu-skeleton"
          data-sidebar="menu-skeleton"
          className={cn("flex h-8 items-center gap-2 rounded-md px-2", className)}
          {...props}>
          {showIcon && (
            <Skeleton className="size-4 rounded-md" data-sidebar="menu-skeleton-icon" />
          )}
          <Skeleton
            className="h-4 max-w-(--skeleton-width) flex-1"
            data-sidebar="menu-skeleton-text"
            style={
              {
                "--skeleton-width": width
              }
            } />
        </div>)
      );
    }

```

```

function SidebarMenuSub({
  className,
  ...props
}) {

```

```

return (
  (<ul
    data-slot="sidebar-menu-sub"
    data-sidebar="menu-sub"
    className={cn(
      "border-sidebar-border mx-3.5 flex min-w-0 translate-x-px flex-col gap-1 border-l px-
2.5 py-0.5",
      "group-data-[collapsible=icon]:hidden",
      className
    )}
    {...props} />)
  );
}

```

```

function SidebarMenuSubItem({
  className,
  ...props
}) {
  return (
    (<li
      data-slot="sidebar-menu-sub-item"
      data-sidebar="menu-sub-item"
      className={cn("group/menu-sub-item relative", className)}
      {...props} />)
    );
}

```

```

function SidebarMenuSubButton({
  asChild = false,
  size = "md",
  isActive = false,

```

```

className,

...props
}) {

const Comp = asChild ? Slot : "a"

return (
  (<Comp
    data-slot="sidebar-menu-sub-button"
    data-sidebar="menu-sub-button"
    data-size={size}
    data-active={isActive}
    className={cn(
      "text-sidebar-foreground ring-sidebar-ring hover:bg-sidebar-accent hover:text-sidebar-
accent-foreground active:bg-sidebar-accent active:text-sidebar-accent-foreground
[&>svg]:text-sidebar-accent-foreground flex h-7 min-w-0 -translate-x-px items-center gap-2
overflow-hidden rounded-md px-2 outline-hidden focus-visible:ring-2 disabled:pointer-
events-none disabled:opacity-50 aria-disabled:pointer-events-none aria-disabled:opacity-50
[&>span:last-child]:truncate [&>svg]:size-4 [&>svg]:shrink-0",
      "data-[active=true]:bg-sidebar-accent data-[active=true]:text-sidebar-accent-
foreground",
      size === "sm" && "text-xs",
      size === "md" && "text-sm",
      "group-data-[collapsible=icon]:hidden",
      className
    )}
  )}
  {...props} />)
);
}

export {
  Sidebar,
  SidebarContent,
  SidebarFooter,

```

SidebarGroup,  
SidebarGroupAction,  
SidebarGroupContent,  
SidebarGroupLabel,  
SidebarHeader,  
SidebarInput,  
SidebarInset,  
SidebarMenu,  
SidebarMenuAction,  
SidebarMenuBadge,  
SidebarMenuButton,  
SidebarMenuItem,  
SidebarMenuSkeleton,  
SidebarMenuSub,  
SidebarMenuSubButton,  
SidebarMenuSubItem,  
SidebarProvider,  
SidebarRail,  
SidebarSeparator,  
SidebarTrigger,  
useSidebar,  
}

### **src\components\ui\Skeleton.jsx:-**

```
import { cn } from "@lib/Utils"
```

```
function Skeleton({
  className,
  ...props
}) {
  return (
    <div
      data-slot="skeleton"
      className={cn("bg-accent animate-pulse rounded-md", className)}
      {...props} />
    );
}
```

```
export { Skeleton }
```

### **src\components\ui\tooltip.jsx:-**

```
"use client"
```

```
import * as React from "react"
```

```
import * as TooltipPrimitive from "@radix-ui/react-tooltip"
```

```
import { cn } from "@lib/Utils"
```

```
function TooltipProvider({
  delayDuration = 0,
  ...props
}) {
  return (
    <TooltipPrimitive.Provider data-slot="tooltip-provider"
      delayDuration={delayDuration} {...props} />
  );
}
```



```
function Tooltip({
  ...props
}) {
  return (
    (<TooltipProvider>
      <TooltipPrimitive.Root data-slot="tooltip" {...props} />
    </TooltipProvider>)
  );
}
```

```
function TooltipTrigger({
  ...props
}) {
  return <TooltipPrimitive.Trigger data-slot="tooltip-trigger" {...props} />;
}
```

```
function TooltipContent({
  className,
  sideOffset = 0,
  children,
  ...props
}) {
  return (
    (<TooltipPrimitive.Portal>
      <TooltipPrimitive.Content
        data-slot="tooltip-content"
        sideOffset={sideOffset}
        className={cn(
          "bg-primary text-primary-foreground animate-in fade-in-0 zoom-in-95 data-
[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=closed]:zoom-out-95
data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-
```

```
[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2 z-50 w-fit origin(--radix-tooltip-content-transform-origin) rounded-md px-3 py-1.5 text-xs text-balance",
```

```

    className
  )}
  {...props}>
  {children}
  <TooltipPrimitive.Arrow
    className="bg-primary fill-primary z-50 size-2.5 translate-y-[calc(-50%_-_2px)]
rotate-45 rounded-[2px]" />
  </TooltipPrimitive.Content>
</TooltipPrimitive.Portal>)
);
}

```

```
export { Tooltip, TooltipTrigger, TooltipContent, TooltipProvider }
```

**src\components\app-sidebar.jsx:-**

```
import { Sidebar } from "@components/ui/sidebar";
```

```
import { Link } from "react-router-dom";
```

```

export function AppSidebar() {
  return (
    <Sidebar>
      <div className="p-4">
        <ul>
          <li className="text-xl font-bold color-aqua"><Link to="/books">Books</Link></li>
          <li className="text-xl font-bold color-aqua"><Link
to="/wishlist">Wishlist</Link></li>
        </ul>
      </div>
    </Sidebar>
  );
}

```

**src\components\BookCard.jsx:-**

```
import { Button } from "@components/ui/button"
import {
  Card,
  CardContent,
  CardDescription,
  CardFooter,
  CardHeader,
  CardTitle,
} from "@components/ui/card"
import { addToWishlist } from "@lib/axios"
import { useState } from "react"

export function BookCard({ book, inWishlist }) {

  const [isLoading, setIsLoading] = useState(false)

  const handleAddToWishlist = async () => {
    try {
      setIsLoading(true)
      await addToWishlist(book._id)
    } catch (error) {
      console.error('Error adding to wishlist:', error)
    } finally {
      setIsLoading(false)
    }
  }

  return (
    <Card className="w-[350px]">
      <CardHeader>
```

```

<CardTitle>{book.title}</CardTitle>

<CardDescription>By {book.author}</CardDescription>

</CardHeader>

<CardContent>

  {book.image && (
    <div className="w-full h-56 flex items-center justify-center overflow-hidden
rounded-lg">
      <img
        src={book.image}
        alt={book.title}
        className="w-full h-full object-contain rounded-lg"
      />
    </div>
  )}

  <p className="text-sm text-gray-600 mt-2">{book.description}</p>

  <div className="mt-1">
    <span className="text-sm font-semibold">Published:</span> {book.publishedYear}
  </div>

  <div className="mt-1 flex flex-wrap gap-1">
    {book.genre.map((g, index) => (
      <span
        key={index}
        className="px-2 py-1 bg-primary/10 text-primary text-xs rounded-full"
      >
        {g}
      </span>
    ))}
  </div>

</CardContent>

<CardFooter>

  <div className="grid grid-cols-2 gap-2">

```

```

    {!inWishlist && (
      <Button
        onClick={handleAddToWishlist}
        disabled={isLoading}
        className="w-full"
      >
        {isLoading ? "Adding..." : "Add to Wishlist"}
      </Button>
    )}
    {inWishlist && (
      <Button
        variant="secondary"
        disabled
      >
        In Wishlist
      </Button>)}
    <Button><a href={book.forwardedTo} target="_blank">buy it</a></Button>
  </div>
</CardFooter>
</Card>
)
}

```

#### **src\components\CreateBookForm.jsx:-**

```

import { useState } from "react";
import { Input } from "@components/ui/input";
import { Button } from "@components/ui/button";
import { createBook } from "@lib/axios";

export function CreateBookForm({ onSuccess }) {
  const [formData, setFormData] = useState({
    title: "",

```

```

image: "",
author: "",
description: "",
publishedYear: "",
genre: "",
forwardedTo: ""
});

const [isLoading, setIsLoading] = useState(false);

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    setIsLoading(true);
    // Convert genre string to array
    const bookData = {
      ...formData,
      genre: formData.genre.split(",").map((g) => g.trim()),
      publishedYear: parseInt(formData.publishedYear)
    };
    await createBook(bookData);
    setFormData({
      title: "",
      image: "",
      author: "",
      description: "",
      publishedYear: "",
      genre: "",
      forwardedTo: ""
    });
    if (onSuccess) onSuccess();
  } catch (error) {

```

```

    console.error("Error creating book:", error);
  } finally {
    setIsLoading(false);
  }
};

const handleChange = (e) => {
  const { name, value } = e.target;
  setFormData((prev) => ({
    ...prev,
    [name]: value
  }));
};

return (
  <form onSubmit={handleSubmit} className="space-y-4 w-full max-w-md">
    <div>
      <Input name="title" placeholder="Book Title" value={formData.title}
onChange={handleChange} required />
    </div>
    <div>
      <Input name="image" placeholder="Image URL" value={formData.image}
onChange={handleChange} required />
    </div>
    <div>
      <Input name="author" placeholder="Author" value={formData.author}
onChange={handleChange} required />
    </div>
    <div>
      <Input name="description" placeholder="Description" value={formData.description}
onChange={handleChange} required />
    </div>
  </form>
);

```

```

    <div>

      <Input name="publishedYear" type="number" placeholder="Published Year"
value={formData.publishedYear} onChange={handleChange} required />

    </div>

    <div>

      <Input name="genre" placeholder="Genres (comma-separated)"
value={formData.genre} onChange={handleChange} required />

    </div>

    <div>

      <Input name="forwardedTo" placeholder="Forwarded To URL"
value={formData.forwardedTo} onChange={handleChange} required />

    </div>

    <Button type="submit" className="w-full" disabled={isLoading}>

      {isLoading ? "Creating..." : "Create Book"}

    </Button>

  </form>

);
}

```

#### **src\components\layout.jsx:-**

```

import { SidebarProvider, SidebarTrigger } from "../ui/sidebar";
import { AppSidebar } from "../app-sidebar";

export default function Layout({ children }) {
  return (
    <SidebarProvider>

      { /* Sidebar Component */ }

      <AppSidebar />

      { /* Main Content */ }

      <main className="flex flex-col w-full">

        { /* Sidebar Toggle Button */ }

        <SidebarTrigger>

```



```

    </SidebarTrigger>

    {children}
  </main>
</SidebarProvider>
);
}

```

#### **src\components\hooks\use-mobile.js:-**

```
import * as React from "react"
```

```
const MOBILE_BREAKPOINT = 768
```

```
export function useIsMobile() {
```

```
  const [isMobile, setIsMobile] = React.useState(undefined)
```

```
  React.useEffect(() => {
```

```
    const mql = window.matchMedia(`(max-width: ${MOBILE_BREAKPOINT - 1}px)`)
```

```
    const onChange = () => {
```

```
      setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
```

```
    }
```

```
    mql.addEventListener("change", onChange)
```

```
    setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
```

```
    return () => mql.removeEventListener("change", onChange);
```

```
  }, [])
```

```
  return !!isMobile
```

```
}
```

#### **src\pages\Auth.jsx:-**

```
import { useState } from "react"
```

```
import { useNavigate } from "react-router-dom"
```

```

import { Button } from "@components/ui/button"
import { Input } from "@components/ui/input"
import { login, signup } from "@lib/axios"
import { useDispatch } from "react-redux"
import { login as authLogin } from "../store/authSlice";

export function AuthPage() {
  const [isLogin, setIsLogin] = useState(true)
  const [formData, setFormData] = useState({
    email: "",
    password: "",
    name: ""
  })
  const [loading, setLoading] = useState(false)
  const navigate = useNavigate()
  const dispatch = useDispatch()

  const handleSubmit = async (e) => {
    e.preventDefault()
    let response;
    try {
      setLoading(true)
      if (isLogin) {
        response = await login(formData.email, formData.password)
      } else {
        response = await signup(formData.email, formData.password, formData.name)
      }
    }

    dispatch(authLogin(response.data))

    navigate("/books")
  }
}

```

```

    } catch (error) {
      console.error('Authentication error:', error)
    } finally {
      setLoading(false)
    }
  }
}

```

```

const handleChange = (e) => {
  const { name, value } = e.target
  setFormData(prev => ({
    ...prev,
    [name]: value
  })))
}

```

```

return (
  <div className="min-h-screen flex items-center justify-center bg-gray-50">
    <div className="w-full max-w-md space-y-8 p-8 bg-white rounded-xl shadow-lg">
      <div className="text-center">
        <h2 className="text-3xl font-bold">{isLoggedIn ? "BookSpire Login" : "BookSpire Sign
Up"}</h2>
        <p className="mt-2 text-gray-600">
          {isLoggedIn ? "Welcome back!" : "Create your account"}
        </p>
      </div>
      <form onSubmit={handleSubmit} className="mt-8 space-y-6">
        {!isLoggedIn && (
          <div>
            <Input
              name="name"
              type="text"

```

```

        placeholder="Full Name"
        value={formData.name}
        onChange={handleChange}
        required={!isLogin}
      />
    </div>
  )}
</div>
  <Input
    name="email"
    type="email"
    placeholder="Email address"
    value={formData.email}
    onChange={handleChange}
    required
  />
</div>
<div>
  <Input
    name="password"
    type="password"
    placeholder="Password"
    value={formData.password}
    onChange={handleChange}
    required
  />
</div>
  <Button type="submit" className="w-full" disabled={loading}>
    {loading ? "Processing..." : isLogin ? "Login" : "Sign Up"}
  </Button>
</form>

```

```

<div className="text-center mt-4">
  <button
    type="button"
    onClick={() => setIsLogin(!isLogin)}
    className="text-sm text-primary hover:underline"
  >
    {isLogin ? "Need an account? Sign up" : "Already have an account? Login"}
  </button>
</div>
</div>
</div>
)
}

```

#### **src\pages\Books.jsx:-**

```

import { useEffect, useState } from "react"
import { getAllBooks } from "@lib/axios"
import { BookCard } from "@components/BookCard"
import { CreateBookForm } from "@components/CreateBookForm"
export function BooksPage() {
  const [books, setBooks] = useState([])
  const [loading, setLoading] = useState(true)
  const [showCreateForm, setShowCreateForm] = useState(false)
  const fetchBooks = async () => {
    try {
      setLoading(true)
      const response = await getAllBooks()
      setBooks(response.data)
    } catch (error) {
      console.error('Error fetching books:', error)
    } finally {
      setLoading(false)
    }
  }
}

```

```

    }
  }
  useEffect(() => {
    fetchBooks()
  }, [])
  if (loading) {
    return <div className="flex justify-center items-center min-h-screen">Loading...</div>
  }
  return (
    <div className="container py-8 px-4">
      <div className="flex justify-between items-center mb-8">
        <h1 className="text-3xl font-bold">Books</h1>
        <button
          onClick={() => setShowCreateForm(!showCreateForm)}
          className="bg-primary text-white px-4 py-2 rounded-md"
        >
          {showCreateForm ? "Hide Form" : "Add New Book"}
        </button>
      </div>
      {showCreateForm && (
        <div className="mb-8">
          <CreateBookForm onSuccess={() => {
            fetchBooks()
            setShowCreateForm(false)
          }} />
        </div>
      )}
    </div>
    <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
      {books.map((book) => (
        <BookCard key={book._id} book={book} />
      ))}
    </div>
  )
}

```

```

    ))}
  </div>
</div>
)
}

```

#### **src\pages\Wishlist.jsx:-**

```

import { useEffect, useState } from "react"
import { getWishlist } from "@lib/axios"
import { BookCard } from "@components/BookCard"

export function WishlistPage() {
  const [wishlist, setWishlist] = useState([])
  const [loading, setLoading] = useState(true)

  useEffect(() => {
    const fetchWishlist = async () => {
      try {
        setLoading(true)
        const response = await getWishlist()
        setWishlist(response.data.books)
      } catch (error) {
        console.error('Error fetching wishlist:', error)
      } finally {
        setLoading(false)
      }
    }
    fetchWishlist()
  }, [])

  if (loading) {
    return <div className="flex justify-center items-center min-h-screen">Loading...</div>
  }
}

```

```

return (
  <div className="container py-8 px-4">
    <h1 className="text-3xl font-bold mb-8">My Wishlist</h1>
    {wishlist.length === 0 ? (
      <p className="text-center text-gray-500">Your wishlist is empty</p>
    ) : (
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
        {wishlist.map((book) => (
          <BookCard key={book._id} book={book} inWishlist={true} />
        ))}
      </div>
    )}
  </div>
)
}

```

#### **src/store/authSlice.js:-**

```
import { createSlice } from "@reduxjs/toolkit";
```

```

const initialState = {
  status : false,
  userData: null
}

```

```

const authSlice = createSlice({
  name: "auth",
  initialState,
  reducers: {
    login: (state, action) => {
      state.status = true;
      let temp = action.payload
      if(temp){

```



```

        state.userData = temp
      }
    },
    logout: (state) => {
      state.status = false;
      state.userData = null;
    }
  }
}
}))

```

```
export const {login, logout } = authSlice.actions;
```

```
export default authSlice.reducer;
```

**src/store/store.js:-**

```
import {configureStore} from '@reduxjs/toolkit';
```

```
import authSlice from './authSlice.js';
```

```
const store = configureStore({
```

```
  reducer: {
```

```
    auth : authSlice,
```

```
  }
```

```
});
```

```
export default store;
```

**src\App.jsx:-**

```
import { BrowserRouter as Router, Routes, Route, Navigate } from "react-router-dom";
```

```
import { AuthPage } from "../pages/Auth.jsx";
```

```
import { BooksPage } from "../pages/Books.jsx";
```

```
import { WishlistPage } from "../pages/Wishlist.jsx";
```

```
import Auth from "../lib/Auth.jsx";
```

```

import { useEffect } from "react";
import { getCurrentUser } from "../lib/axios.js";
import { useDispatch } from "react-redux";
import { login as authLogin } from "../store/authSlice.js";
import Layout from "../components/Layout"; // Import the Layout component

function App() {
  const dispatch = useDispatch();

  useEffect(() => {
    const fetchDetails = async () => {
      const response = await getCurrentUser();
      dispatch(authLogin(response.data));
    };

    fetchDetails();
  }, []);

  return (
    <Router>
    <Routes>
      <Route path="/" element={<Navigate to="/auth" replace />} />
      <Route path="/auth" element={<Auth authentication={false}><AuthPage /></Auth>} />
      <Route path="/books" element={<Layout><Auth><BooksPage /></Auth></Layout>} />
      <Route path="/wishlist" element={<Layout><Auth><WishlistPage
/></Auth></Layout>} />
    </Routes>
    </Router>
  );
}
export default App;

```

## src\index.css:-

```
@import "tailwindcss";
@import "tw-animate-css";

@custom-variant dark (&:is(.dark *));

:root {
  --radius: 0.625rem;
  --background: oklch(1 0 0);
  --foreground: oklch(0.145 0 0);
  --card: oklch(1 0 0);
  --card-foreground: oklch(0.145 0 0);
  --popover: oklch(1 0 0);
  --popover-foreground: oklch(0.145 0 0);
  --primary: oklch(0.205 0 0);
  --primary-foreground: oklch(0.985 0 0);
  --secondary: oklch(0.97 0 0);
  --secondary-foreground: oklch(0.205 0 0);
  --muted: oklch(0.97 0 0);
  --muted-foreground: oklch(0.556 0 0);
  --accent: oklch(0.97 0 0);
  --accent-foreground: oklch(0.205 0 0);
  --destructive: oklch(0.577 0.245 27.325);
  --border: oklch(0.922 0 0);
  --input: oklch(0.922 0 0);
  --ring: oklch(0.708 0 0);
  --chart-1: oklch(0.646 0.222 41.116);
  --chart-2: oklch(0.6 0.118 184.704);
  --chart-3: oklch(0.398 0.07 227.392);
  --chart-4: oklch(0.828 0.189 84.429);
  --chart-5: oklch(0.769 0.188 70.08);
```

```
--sidebar: oklch(0.985 0 0);
--sidebar-foreground: oklch(0.145 0 0);
--sidebar-primary: oklch(0.205 0 0);
--sidebar-primary-foreground: oklch(0.985 0 0);
--sidebar-accent: oklch(0.97 0 0);
--sidebar-accent-foreground: oklch(0.205 0 0);
--sidebar-border: oklch(0.922 0 0);
--sidebar-ring: oklch(0.708 0 0);
}
```

```
.dark {
  --background: oklch(0.145 0 0);
  --foreground: oklch(0.985 0 0);
  --card: oklch(0.205 0 0);
  --card-foreground: oklch(0.985 0 0);
  --popover: oklch(0.205 0 0);
  --popover-foreground: oklch(0.985 0 0);
  --primary: oklch(0.922 0 0);
  --primary-foreground: oklch(0.205 0 0);
  --secondary: oklch(0.269 0 0);
  --secondary-foreground: oklch(0.985 0 0);
  --muted: oklch(0.269 0 0);
  --muted-foreground: oklch(0.708 0 0);
  --accent: oklch(0.269 0 0);
  --accent-foreground: oklch(0.985 0 0);
  --destructive: oklch(0.704 0.191 22.216);
  --border: oklch(1 0 0 / 10%);
  --input: oklch(1 0 0 / 15%);
  --ring: oklch(0.556 0 0);
  --chart-1: oklch(0.488 0.243 264.376);
  --chart-2: oklch(0.696 0.17 162.48);
}
```

```
--chart-3: oklch(0.769 0.188 70.08);
--chart-4: oklch(0.627 0.265 303.9);
--chart-5: oklch(0.645 0.246 16.439);
--sidebar: oklch(0.205 0 0);
--sidebar-foreground: oklch(0.985 0 0);
--sidebar-primary: oklch(0.488 0.243 264.376);
--sidebar-primary-foreground: oklch(0.985 0 0);
--sidebar-accent: oklch(0.269 0 0);
--sidebar-accent-foreground: oklch(0.985 0 0);
--sidebar-border: oklch(1 0 0 / 10%);
--sidebar-ring: oklch(0.556 0 0);
}
```

```
@theme inline {
  --radius-sm: calc(var(--radius) - 4px);
  --radius-md: calc(var(--radius) - 2px);
  --radius-lg: var(--radius);
  --radius-xl: calc(var(--radius) + 4px);
  --color-background: var(--background);
  --color-foreground: var(--foreground);
  --color-card: var(--card);
  --color-card-foreground: var(--card-foreground);
  --color-popover: var(--popover);
  --color-popover-foreground: var(--popover-foreground);
  --color-primary: var(--primary);
  --color-primary-foreground: var(--primary-foreground);
  --color-secondary: var(--secondary);
  --color-secondary-foreground: var(--secondary-foreground);
  --color-muted: var(--muted);
  --color-muted-foreground: var(--muted-foreground);
  --color-accent: var(--accent);
```

```

--color-accent-foreground: var(--accent-foreground);
--color-destructive: var(--destructive);
--color-border: var(--border);
--color-input: var(--input);
--color-ring: var(--ring);
--color-chart-1: var(--chart-1);
--color-chart-2: var(--chart-2);
--color-chart-3: var(--chart-3);
--color-chart-4: var(--chart-4);
--color-chart-5: var(--chart-5);
--color-sidebar: var(--sidebar);
--color-sidebar-foreground: var(--sidebar-foreground);
--color-sidebar-primary: var(--sidebar-primary);
--color-sidebar-primary-foreground: var(--sidebar-primary-foreground);
--color-sidebar-accent: var(--sidebar-accent);
--color-sidebar-accent-foreground: var(--sidebar-accent-foreground);
--color-sidebar-border: var(--sidebar-border);
--color-sidebar-ring: var(--sidebar-ring);
}

```

```

@layer base {
  * {
    @apply border-border outline-ring/50;
  }
  body {
    @apply bg-background text-foreground;
  }
}

```

## **Backend:-**

### **src\config\database.js:-**

```
import mongoose from 'mongoose';
import dotenv from 'dotenv';
dotenv.config();
const connectDB = async () => {
  try {
    const conn = await mongoose.connect(process.env.MONGODB_URI, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log(`MongoDB Connected: ${conn.connection.host}`);
  } catch (error) {
    console.error(`Error: ${error.message}`);
    process.exit(1);
  }
};
export default connectDB;
```

### **src\controllers\auth.controller.js:-**

```
import jwt from 'jsonwebtoken';
import { createResponse } from '../utils/response.js';
import User from '../models/user.model.js';
export const signup = async (req, res) => {
  try {
    const { email, password, name } = req.body;

    const existingUser = await User.findOne({ email });
    if (existingUser) {
      return res.status(400).json(createResponse(false, 'User already exists'));
    }
  }
}
```

```

const user = await User.create({
  email,
  password,
  name
});

const token = jwt.sign({ userId: user._id }, process.env.JWT_SECRET || 'your-secret-key',
{ expiresIn: '1h' });

res.cookie('token', token, { httpOnly: true });

res.status(200).json(createResponse(true, 'User created successfully', { email: user.email,
name: user.name }));
} catch (error) {
  res.status(500).json(createResponse(false, 'Error creating user'));
}
};

export const login = async (req, res) => {
  try {
    const { email, password } = req.body;
    const user = await User.findOne({ email });

    if (!user || !password) {
      return res.status(401).json(createResponse(false, 'Invalid credentials'));
    }

    const passwordMatch = await user.comparePassword(password);
    console.log(passwordMatch);

    if (!passwordMatch) {
      return res.status(401).json(createResponse(false, 'Invalid credentials'));
    }
  }
}

```



```
const token = jwt.sign({ userId: user._id }, process.env.JWT_SECRET || 'your-secret-key',
{ expiresIn: '1h' });
```

```
res.cookie('token', token, { httpOnly: true });
res.json(createResponse(true, 'Login successful', { email: user.email, name: user.name }));
} catch (error) {
res.status(500).json(createResponse(false, 'Error during login'));
}
};
```

```
export const logout = (req, res) => {
res.clearCookie('token');
res.json(createResponse(true, 'Logged out successfully'));
};
```

```
export const getCurrentUser = (req, res) => {
res.json(createResponse(true, 'User found', req.user));
}
```

**src\controllers\book.controller.js:-**

```
import Book from '../models/book.model.js';
import Wishlist from '../models/wishlist.model.js';
import { createResponse } from '../utils/response.js';
```

```
export const getAllBooks = async (req, res) => {
try {
const books = await Book.find().select('-__v');
res.json(createResponse(true, 'Books retrieved successfully', books));
} catch (error) {
res.status(500).json(createResponse(false, 'Error retrieving books'));
}
};
```

```

export const addToWishlist = async (req, res) => {
  try {
    const { bookId } = req.body;
    const userId = req.user.userId;

    // Verify book exists
    const book = await Book.findById(bookId);
    if (!book) {
      return res.status(404).json(createResponse(false, 'Book not found'));
    }

    // Find or create wishlist for user
    let wishlist = await Wishlist.findOne({ user: userId });
    if (!wishlist) {
      wishlist = new Wishlist({ user: userId, books: [] });
    }

    // Add book if not already in wishlist
    if (!wishlist.books.includes(bookId)) {
      wishlist.books.push(bookId);
      await wishlist.save();
    }

    // Populate books details
    await wishlist.populate('books', '-__v');

    res.json(createResponse(true, 'Book added to wishlist', wishlist));
  } catch (error) {
    res.status(500).json(createResponse(false, 'Error adding book to wishlist'));
  }
};

```

```

export const getWishlist = async (req, res) => {
  try {
    const userId = req.user.userId;
    const wishlist = await Wishlist.findOne({ user: userId })
      .populate('books', '-__v');

    if (!wishlist) {
      return res.json(createResponse(true, 'Wishlist is empty', { books: [] }));
    }

    res.json(createResponse(true, 'Wishlist retrieved successfully', wishlist));
  } catch (error) {
    res.status(500).json(createResponse(false, 'Error retrieving wishlist'));
  }
};

export const createBook = async (req, res) => {
  try {
    const { title, image, author, description, publishedYear, genre, forwardedTo } = req.body;
    if (!title || !image || !author || !description || !publishedYear || !genre || !forwardedTo) {
      return res.status(400).json(createResponse(false, 'Missing required fields'));
    }

    // Create a new book entry
    console.log(1);

    const newBook = await Book.create({
      title,
      image,
      author,
      description,

```

```

    publishedYear,
    genre,
    forwardedTo
  });

  res.status(201).json(createResponse(true, 'Book created successfully', newBook));
} catch (error) {
  console.log(error);

  res.status(500).json(createResponse(false, 'Error creating book', error.message));
}
};

src\middleware\auth.middleware.js:-
import jwt from 'jsonwebtoken';
import { createResponse } from '../utils/response.js';

export const validateAuth = (req, res, next) => {
  const { email, password } = req.body;

  if (!email || !password) {
    return res.status(400).json(
      createResponse(false, 'Email and password are required')
    );
  }

  if (password.length < 6) {
    return res.status(400).json(
      createResponse(false, 'Password must be at least 6 characters long')
    );
  }
}

```

```

    next();
  };

export const authenticateToken = (req, res, next) => {
  const token = req.cookies.token;

  if (!token) {
    return res.status(401).json(
      createResponse(false, 'Authentication required')
    );
  }

  try {
    const decoded = jwt.verify(token, process.env.JWT_SECRET || 'your-secret-key');
    req.user = decoded;
    next();
  } catch (error) {
    res.status(401).json(
      createResponse(false, 'Invalid or expired token')
    );
  }
};

```

**src\middleware\error.middleware.js:-**

```

export const errorHandler = (err, req, res, next) => {
  console.error(err.stack);

  res.status(500).json({
    success: false,
    message: 'Something went wrong!',
    error: process.env.NODE_ENV === 'development' ? err.message : undefined
  });
};

```

```
};
```

**src\models\book.model.js:-**

```
import mongoose from 'mongoose';
```

```
const bookSchema = new mongoose.Schema({
```

```
  title: {
```

```
    type: String,
```

```
    required: true,
```

```
    trim: true
```

```
  },
```

```
  image : {
```

```
    type: String,
```

```
    trim: true,
```

```
    required: true
```

```
  },
```

```
  forwardedTo: {
```

```
    type: String,
```

```
    trim: true
```

```
  },
```

```
  author: {
```

```
    type: String,
```

```
    required: true,
```

```
    trim: true
```

```
  },
```

```
  description: {
```

```
    type: String,
```

```
    trim: true
```

```
  },
```

```
  publishedYear: {
```

```
    type: Number
```

```
  },
```

```
genre: [{
  type: String,
  trim: true
}],
createdAt: {
  type: Date,
  default: Date.now
}
});
```

```
const Book = mongoose.model('Book', bookSchema);
```

```
export default Book;
```

**src\models\user.model.js:-**

```
import mongoose from 'mongoose';
```

```
import bcrypt from 'bcryptjs';
```

```
const userSchema = new mongoose.Schema({
```

```
  email: {
    type: String,
    required: true,
    unique: true,
    trim: true,
    lowercase: true
  },
```

```
  password: {
    type: String,
    required: true,
    minlength: 6
  },
```

```
  name: {
```

```

    type: String,
    trim: true
  },
  createdAt: {
    type: Date,
    default: Date.now
  }
});

// Hash password before saving
userSchema.pre('save', async function(next) {
  if (!this.isModified('password')) return next();
  this.password = await bcrypt.hash(this.password, 10);
});

// Method to compare password
userSchema.methods.comparePassword = async function(candidatePassword) {
  return bcrypt.compare(candidatePassword, this.password);
};

const User = mongoose.model('User', userSchema);

export default User;

```

**src\models\wishlist.model.js:-**

```
import mongoose from 'mongoose';
```



```

const wishlistSchema = new mongoose.Schema({
  user: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'User',
    required: true
  },
  books: [{
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Book'
  }],
  createdAt: {
    type: Date,
    default: Date.now
  },
  lastModified: {
    type: Date,
    default: Date.now
  }
});

// Update lastModified timestamp before saving
wishlistSchema.pre('save', function(next) {
  this.lastModified = new Date();
  next();
});

const Wishlist = mongoose.model('Wishlist', wishlistSchema);

export default Wishlist;

```

**src\routes\auth.routes.js:-**

```

import express from 'express';

```

```
import { signup, login, logout , getCurrentUser } from '../controllers/auth.controller.js';  
import { validateAuth , authenticateToken} from '../middleware/auth.middleware.js';
```

```
const router = express.Router();
```

```
router.post('/signup', validateAuth, signup);  
router.post('/login', validateAuth, login);  
router.post('/logout', logout);  
router.get("/get-current-user" , authenticateToken , getCurrentUser);
```

```
export default router;
```

#### **src\routes\book.routes.js:-**

```
import express from 'express';  
  
import { getAllBooks, addToWishlist, getWishlist, createBook } from  
  '../controllers/book.controller.js';  
  
import { authenticateToken } from '../middleware/auth.middleware.js';
```

```
const router = express.Router();
```

```
router.get('/', getAllBooks);  
router.get('/wishlist', authenticateToken, getWishlist);  
router.post('/wishlist', authenticateToken, addToWishlist);  
router.post('/create-book', authenticateToken, createBook);
```

```
export default router;
```

#### **src\index.js:-**

```
import express from 'express';
```

```

import cors from 'cors';
import cookieParser from 'cookie-parser';
import dotenv from 'dotenv';
import connectDB from './config/database.js';
import authRoutes from './routes/auth.routes.js';
import bookRoutes from './routes/book.routes.js';
import { errorHandler } from './middleware/error.middleware.js';

dotenv.config();

// Connect to MongoDB
connectDB();

const app = express();

app.use(cors({
  origin: "http://localhost:5173",
  credentials: true,
  allowedHeaders: ['Content-Type', 'Authorization', 'x-csrf-token', 'Access-Control-Allow-Headers', 'Access-Control-Allow-Methods'],
  methods: ['GET', 'POST', 'PATCH', 'DELETE', 'OPTIONS'],
}));

app.use(express.json());
app.use(cookieParser());

// Routes
app.use('/api/auth', authRoutes);
app.use('/api/books', bookRoutes);

// Error handling
app.use(errorHandler);

```

```
const PORT = 3000;  
app.listen(PORT, () => {  
  console.log(`Server is running on port http://localhost${PORT}`);  
});
```

**src\utils\response.js:-**

```
export const createResponse = (success, message, data = null) => {  
  return {  
    success,  
    message,  
    ...(data && { data })  
  };  
};
```

## 4.2) Test Cases

### ✔ Functional Test Cases - Pass Conditions

Test Case ID	Test Scenario	Expected Result	Status
FT-01	Books should be fetched and displayed properly	Books appear with title, author, image, description, published year, and tags	✔ Pass
FT-02	Clicking on "Add to Wishlist" should add the book to the wishlist	The book gets added to the wishlist and persists across sessions	✔ Pass
FT-03	Clicking "buy it" should initiate the purchase process	The system redirects to the checkout or payment page	✔ Pass
FT-04	"Add New Book" button should open a form/modal	Clicking should display a form/modal to enter book details	✔ Pass
FT-05	Book details should be saved correctly upon form submission	Newly added books appear correctly in the book list	✔ Pass
FT-06	Book images should load correctly	All book images should be visible and properly displayed	✔ Pass
FT-07	Wishlist should retain added books on page refresh	The wishlist items persist even after refreshing the page	✔ Pass

### ✔ UI/UX Test Cases - Pass Conditions

Test Case ID	Test Scenario	Expected Result	Status
UX-01	Navigation should be intuitive and easy to use	Users can easily navigate between Books and Wishlist pages without confusion	✔ Pass
UX-02	Book cards should be visually appealing and properly aligned	Each book card is well-structured with proper spacing, images, and readable text	✔ Pass
UX-03	Buttons should be clearly labeled and functional	"Add to Wishlist" and "Buy it" buttons are easily identifiable and work as expected	✔ Pass
UX-04	The "Add New Book" button should be prominent and accessible	Users can quickly locate and interact with the button	✔ Pass
UX-05	Text content should be readable with proper contrast	Font size, color, and contrast are optimized for readability	✔ Pass
UX-06	The interface should be responsive across different screen sizes	The layout adjusts properly on desktop, tablet, and mobile screens	✔ Pass
UX-07	Hover and click effects should provide clear feedback	Buttons and interactive elements provide visual feedback on hover and click	✔ Pass

## **5. Results**

### **Functional Results:**

#### **Book Fetching and Display**

- The system correctly fetches and displays books with title, author, image, description, published year, and tags.
- Books load dynamically without delays.

#### **Add New Book Functionality**

- Clicking the "Add New Book" button opens a form or modal to enter book details.
- The form appears without UI glitches and allows smooth data entry.

#### **Book Image Loading**

- All book images are displayed properly without distortion.
- Images load quickly and adjust well to different screen sizes.

#### **Wishlist Feature**

- Clicking "Add to Wishlist" successfully adds the book to the wishlist.
- Books can be removed from the wishlist without issues.

#### **Buy Button Functionality**

- Clicking the "Buy it" button redirects users to the appropriate purchase page.
- No broken links or incorrect navigation occur.

#### **Navigation and Page Functionality**

- Sidebar navigation between "Books" and "Wishlist" works smoothly.
- No page crashes or broken elements during navigation.

#### **UI/UX Test Cases - Pass Conditions**

#### **Responsive and User-Friendly Design**

- The layout adapts well across different devices (desktop, tablet, and mobile).
- Cards and text are aligned properly for a clean interface.

#### **Typography and Readability**

- Font sizes, colors, and contrast are well-optimized for readability.
- White text on a dark background ensures good visibility.

#### **Interactive Elements and Feedback**

- Buttons respond to clicks with animations and hover effects.
- Clicking buttons provides immediate feedback with no delays.

Form Accessibility

- The "Add New Book" form is easy to navigate and interact with.
- Input fields are labeled clearly, improving usability.

5.1 Output Screen(s):

1) Login Page

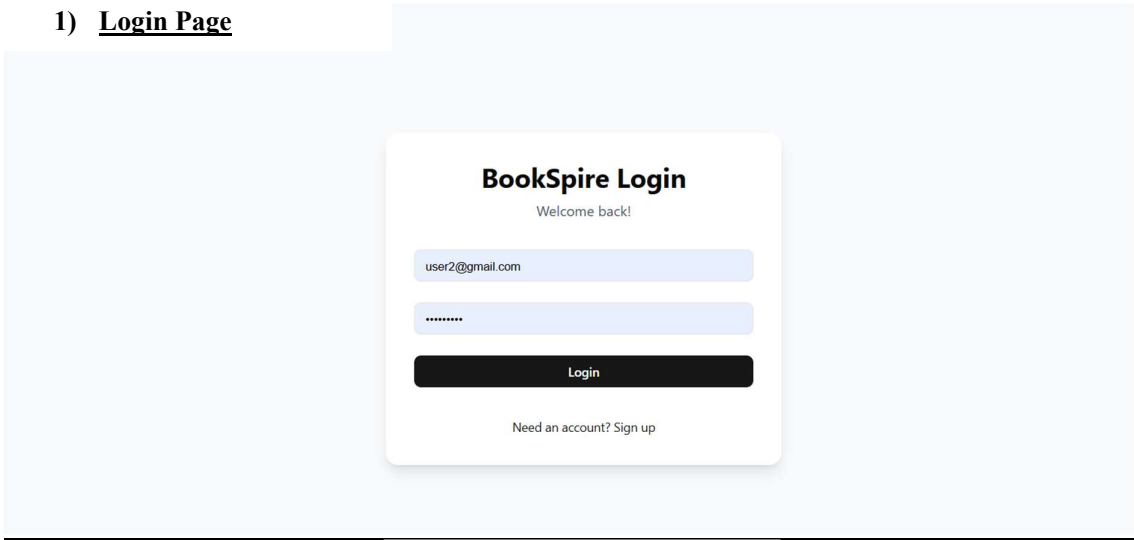


Fig:-5.2.1

2) Index page

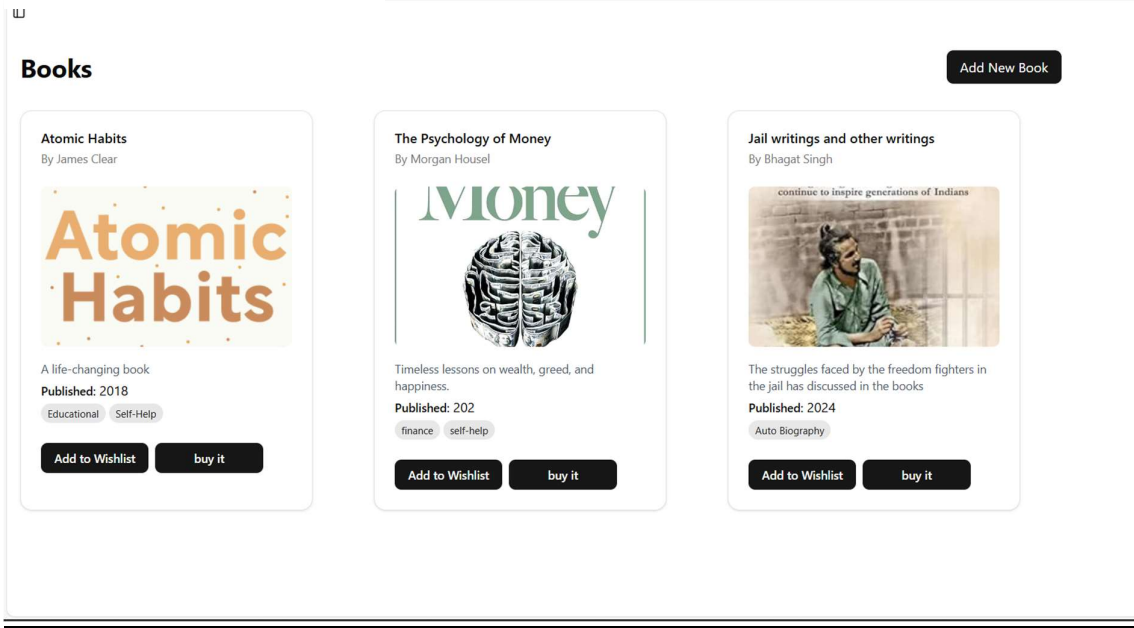


Fig:-5.2.2

### 3) Adding a new book

Books  
Wishlist

Books

To kill a Mocking Bird

[https://m.media-amazon.com/images/I/71e0ZGkR4IL\\_SY466\\_.jpg](https://m.media-amazon.com/images/I/71e0ZGkR4IL_SY466_.jpg)

Harper Lee

a coming-of-age story set in 1930s Alabama, exploring themes of rac

1961

Fiction

<https://www.amazon.in/Kill-Mockingbird-50th-Anniversary/dp/0434C>

Create Book


Fig:-5.2.3

### 4) Updation of the book

Books

Add New Book


Atomic Habits  
By James Clear



A life-changing book  
Published: 2018  
Educational Self-Help

Add to Wishlist buy it


The Psychology of Money  
By Morgan Housel



Timeless lessons on wealth, greed, and happiness.  
Published: 202  
finance self-help

Add to Wishlist buy it

Jail writings and other writings  
By Bhagat Singh



The struggles faced by the freedom fighters in the jail has discussed in the books  
Published: 2024  
Auto Biography

Add to Wishlist buy it

To kill a Mocking Bird  
By Harper Lee

Fig:-5.2.4



## **6. Conclusion**

The BookSpire project effectively integrates real-time book management, user interaction, and a seamless UI/UX experience, making it both engaging and functional.

### **Key Achievements:**

- **Accurate Book Fetching & Display** – The system dynamically retrieves and displays books with essential details such as title, author, image, description, published year, and tags.
- **Wishlist & Purchase Integration** – Users can add books to their wishlist and access purchase options smoothly, enhancing the reading experience.
- **User-Friendly & Responsive UI** – The design is optimized for desktop and mobile, ensuring accessibility and ease of navigation across devices.
- **Performance Stability** – The system efficiently loads book details, images, and interactive elements without any delays or UI glitches.
- **Interactive & Engaging Features** – Buttons respond with hover effects and animations, improving usability and engagement.

The BookSpire project successfully meets its objective of providing an intuitive and feature-rich platform for managing and discovering books. By offering real-time interaction, wishlist tracking, and smooth navigation, the system enhances user engagement. Additionally, the project has allowed the development team to gain valuable expertise in front-end technologies and backend API integration.

With future enhancements, such as advanced search filters, user reviews, and AI-based book recommendations, BookSpire can evolve into a full-scale book management and reading platform, catering to both casual readers and book enthusiasts.

## **7. References**

The development and implementation of the BookSpire project were guided by various technical resources, research articles, and existing book management platforms.

**Below are the key references:**

- **Goodreads** – Book Management & Wishlist Features

**Source:** <https://www.goodreads.com>

Used for understanding book categorization, wishlist functionality, and user interaction features.

- **Amazon Books** – UI/UX Inspiration

**Source:** <https://www.amazon.com/books>

Helped in designing an intuitive interface for book display, purchase options, and user engagement.

- **Open Library API** – Book Data Fetching

**Source:** <https://openlibrary.org/developers/api>

Referenced for book fetching mechanisms and structuring metadata like author details, descriptions, and tags.

- **Mozilla Developer Network (MDN)** – JavaScript & API Integration

**Source:** <https://developer.mozilla.org>

Used for implementing JavaScript event handling, API calls, and dynamic content rendering.

- **W3Schools** – Responsive Web Design & Tailwind CSS

**Source:** <https://www.w3schools.com>

Helped in developing a fully responsive and visually appealing book management interface.