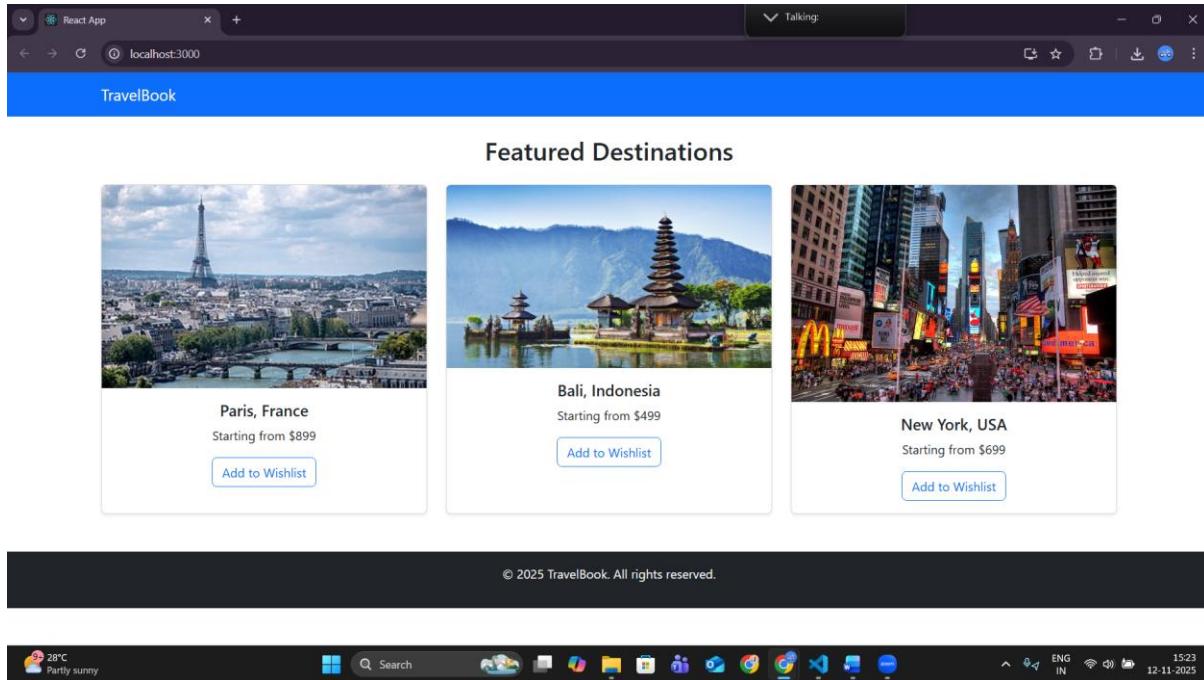


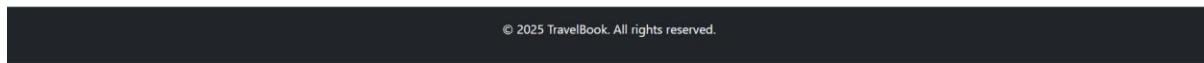
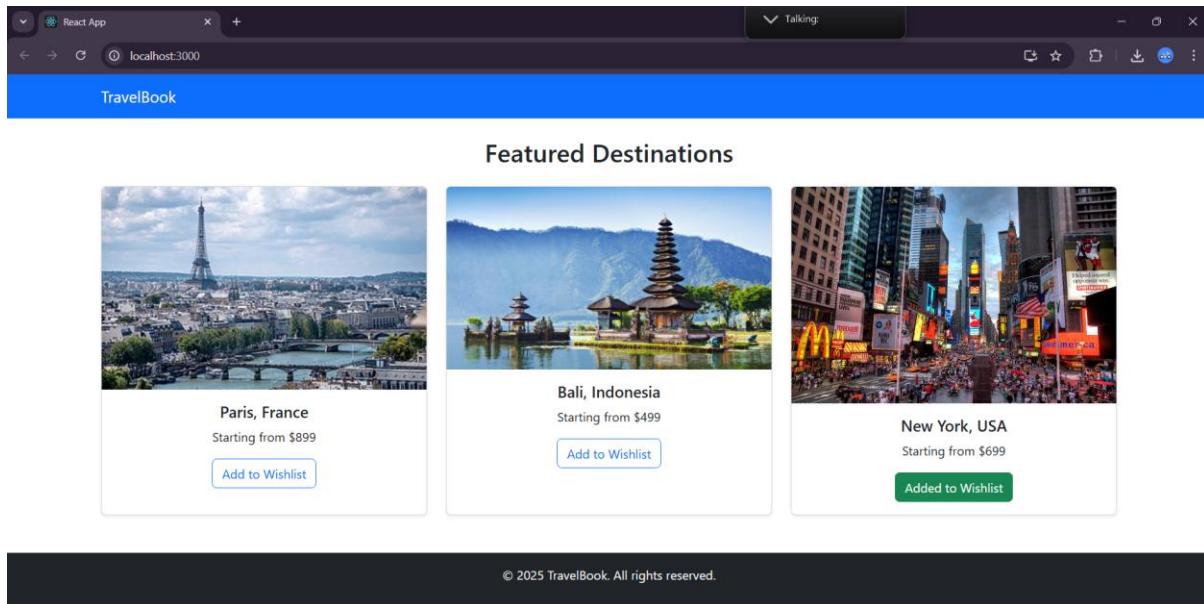
## User Story- 1

- created react app using “npx create-react-app travel-booking”
- Installed bootstrap in travel-booking using “npm install bootstrap”
- imported bootstrap to index file

### Page SS:-



After clicking on added to Wishlist



## file structure and data passing through prop

The screenshot shows a developer's workspace with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS".
  - WIPRO NGA - MERN**: Contains "Mock-1" and "Mock-2" folders, each with "travel-booking" sub-folders containing "node\_modules", "public", and "src" directories.
  - images**: Contains files like "Ball.jpg", "New-York.jpg", "paris.jpg", "#App.css", "App.js", "App.test.js", "index.css", "index.js", and "logo.svg".
- Code Editor:** Displays a component named "App.js" which maps through a list of destinations to render "DestinationCard" components.
- Terminal:** Shows the command "PS C:\Users\Kush\OneDrive\Desktop\Wipro NGA - MERN\Mock-2\travel-booking> npm start" and ESLint errors related to href attributes.
- Status Bar:** Shows "webpack compiled with 1 warning".

## used useState for add to wishlist

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure for "Wipro NGA - MERN". The file "DestinationCard.js" is selected.
- Editor:** Displays the code for "DestinationCard.js". The code uses the `useState` hook from React to manage a wishlist state. It includes a function to handle adding to the wishlist and a button with an `onClick` event handler.
- Terminal:** Shows the command `npm start` being run in the terminal, along with ESLint warnings about anchor validity.
- Bottom Status Bar:** Shows the current file is "DestinationCard.js", the line is 16, column is 27, and the status bar also includes network and battery information.

```
1 import React, { useState } from 'react';
2
3 function DestinationCard({ name, image, price }) {
4   const [wishlist, setWishlist] = useState(false);
5
6   const handleWishlist = () => {
7     setWishlist(!wishlist);
8   };
9
10  return (
11    <div className="col-md-4 mb-4">
12      <div className="card shadow-sm h-100">
13        <img src={image} className="card-img-top" alt={name} />
14        <div className="card-body text-center">
15          <h5 className="card-title">{name}</h5>
16          <p className="card-text">Starting from ${price}</p>
17          <button
18            className="btn ${wishlist ? 'btn-success' : 'btn-outline-primary'}"
19            onClick={handleWishlist}
20          >
21            {wishlist ? 'Added to Wishlist' : 'Add to Wishlist'}
22          </button>
23        </div>
24      </div>
25    </div>
26  );
27}
```

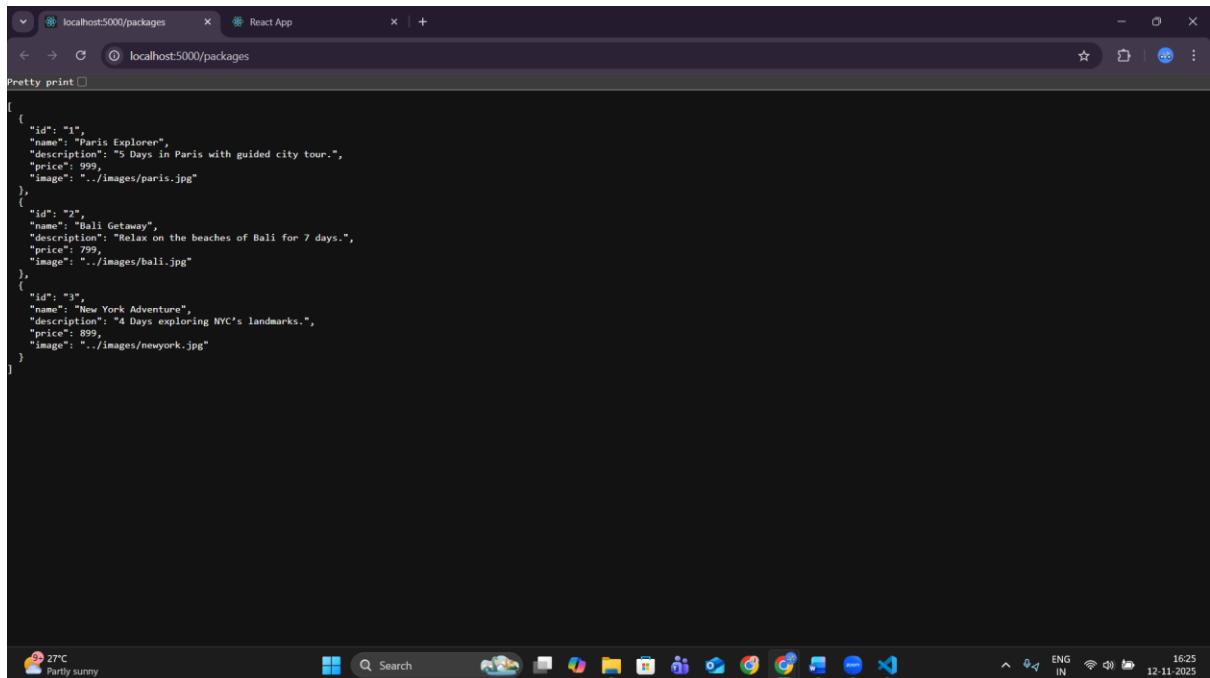
PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Mock-2\travel-booking> npm start  
Provide a valid href, but still need the element to resemble a link, use a button and change it with appropriate styles. Learn more: https://github.com/jsx-eslint/eslint-plugin-jsx-ally/blob/HEAD/docs/rules/anchor-is-valid.md [jsx-ally/anchor-is-valid](#)  
Search for the [keywords](#) to learn more about each warning.  
To ignore, add // eslint-disable-next-line to the line before.

WARNING in [eslint]  
src/components/Header.js  
Line 7:9: The href attribute requires a valid value to be accessible. Provide a valid, navigable address as the href value. If you cannot provide a valid href, but still need the element to resemble a link, use a button and change it with appropriate styles. Learn more: https://github.com/jsx-eslint/eslint-plugin-jsx-ally/blob/HEAD/docs/rules/anchor-is-valid.md [jsx-ally/anchor-is-valid](#)

## User Story -2

- install reactrouter-dom, prop-types using “npm install react-router-dom framer-motion prop-types”

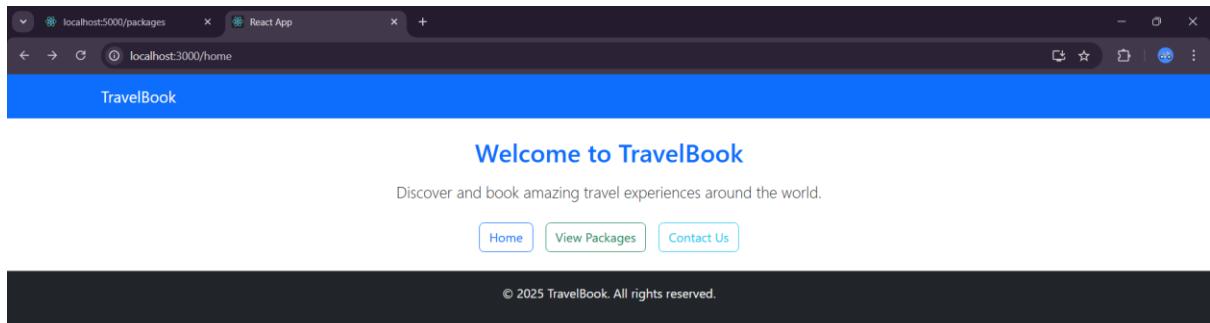
## server output



```
[{"id": "1", "name": "Paris Explorer", "description": "5 Days in Paris with guided city tour.", "price": 999, "image": "../images/paris.jpg"}, {"id": "2", "name": "Bali Getaway", "description": "Relax on the beaches of Bali for 7 days.", "price": 799, "image": "../images/bali.jpg"}, {"id": "3", "name": "New York Adventure", "description": "4 Days exploring NYC's landmarks.", "price": 899, "image": "../images/newyork.jpg"}]
```

The screenshot shows a browser window with the URL `localhost:5000/packages`. The page displays a JSON object representing three travel packages. Each package has an ID, name, description, price, and a corresponding image path. The browser interface includes a top navigation bar with tabs for 'localhost:5000/packages' and 'React App', and a bottom taskbar with various icons.

## HomePage



# Welcome to TravelBook

Discover and book amazing travel experiences around the world.

[Home](#) [View Packages](#) [Contact Us](#)

© 2025 TravelBook. All rights reserved.

The screenshot shows a web application titled 'TravelBook'. The header features the title 'Welcome to TravelBook' and a subtext 'Discover and book amazing travel experiences around the world.'. Below the header are three buttons: 'Home', 'View Packages', and 'Contact Us'. At the bottom of the page is a dark footer bar containing the copyright notice '© 2025 TravelBook. All rights reserved.' The browser window has a blue header bar with the URL `localhost:3000/home`.



## /Packages

The screenshot shows a web browser window for 'TravelBook' displaying 'Available Packages'. The page features a blue header bar with the 'TravelBook' logo. Below it, a section titled 'Available Packages' contains three travel packages, each with a thumbnail image of a city skyline at night:

- Paris Explorer**: 5 Days in Paris with guided city tour.  
Price: \$999  
[View Details](#) [Book Package](#)
- Bali Getaway**: Relax on the beaches of Bali for 7 days.  
Price: \$799  
[View Details](#) [Book Package](#)
- New York Adventure**: 4 Days exploring NYC's landmarks.  
Price: \$899  
[View Details](#) [Book Package](#)

At the bottom of the page, there is a dark footer bar with the text '© 2025 TravelBook. All rights reserved.' and a system tray showing weather, date, and time.

## /contact

The screenshot shows a web browser window for 'TravelBook' displaying the 'Contact Us' page. The page has a blue header bar with the 'TravelBook' logo. The main content area features a heading 'Contact Us' and two lines of contact information:

Email: support@travelbook.com  
Phone: +91 9111111111

A green 'Send Message' button is located below the contact info. At the bottom of the page is a dark footer bar with the text '© 2025 TravelBook. All rights reserved.' and a system tray showing weather, date, and time.



### user story -3

-installed formik yup axios

/book – form will open

localhost:5000/packages x React App x +

localhost:3000/book

## Book a Travel Package

Full name

Full name is required

Email

Package

Travel Date

Travel date is required

Guests

Notes

with form inputs validation coming from formic yup

localhost:5000/packages    React App

localhost:3000/book

## Book a Travel Package

Full name

Full name is required

Email

Package

Travel Date

dd-mm-yyyy

November, 2025 ▾ ↑ ↓

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

after submitting the data is stored in api

A screenshot of a web browser window titled "localhost:3000/book". The form fields include: Email (you@example.com), Package (dropdown menu showing "-- Choose a package --"), Travel Date (dd-mm-yyyy), Guests (1), and Notes (Any special requests?). Below the form are two buttons: "Confirm Booking" (blue) and "Reset". A green success message box at the bottom right contains the text "Booking successful! ID: d445". The browser's address bar shows "localhost:3000/book". The operating system taskbar at the bottom displays the date and time as 12-11-2025, 16:46.

A screenshot of a web browser window titled "localhost:5000/packages" and "localhost:5000/bookings". The page displays a JSON object representing a booking. The JSON is:

```
{  
  "id": "d445",  
  "fullName": "Deepak",  
  "email": "kushwahadeepak2508@gmail.com",  
  "packageId": "2",  
  "date": "2025-11-13",  
  "guests": 1,  
  "notes": ""  
}
```

The browser's address bar shows "localhost:5000/bookings". The operating system taskbar at the bottom displays the date and time as 12-11-2025, 16:47.