

PRINTHUB:

Assistant

A Project Report *Submitted by:*

A.JAGAN REDDY – AP23110011664

A.KIRAN – AP23110011126

k.Hanok – AP23110011236

P.Jyoshna – AP23110011145

Under the Supervision of:

Mr. Syed Arshad

Assistant Professor

Department of Computer Science and Engineering

SRM University-AP

In partial fulfilment for the requirements of the

FullStack Development-I(SEC 160) Project

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



SRM
UNIVERSITY AP

— Andhra Pradesh

Introduction:

PrintHub is a web-based platform designed to make printing services fast, simple, and easily accessible. Many students and professionals often wait in long queues at print shops just to get basic services like Xerox, printing, lamination, or ID card printing. This wastes time and creates unnecessary frustration.

PrintHub solves this problem by bringing all print services online. Users can view services, check prices, upload documents, and place orders directly through the app. Once submitted, the order is added to a digital queue, and users can track the status until the print shop completes their request.

The system removes the need for physical waiting and makes the printing process smooth and organized

Scenario-Based Intro :

Imagine you need to print an assignment urgently. You go to a print shop and find a long line of students waiting. The shop owner is overwhelmed, and you are stuck waiting for 30 minutes.

With PrintHub, the experience becomes much easier. You open the app, choose "Document Printing", upload your PDF, select the number of pages and copies, and confirm your order. Immediately, the app shows your queue number and estimated completion time.

Once your order is ready, you get a notification. You simply walk to the shop and pick up your documents without waiting.

This makes print services faster, smarter, and hassle-free.

Problem Statement:

❖ **Traditional print shops face issues like:**

- Long queues and slow service
- Manual handling of orders
- No clear order tracking
- Customers waiting without updates
- Miscommunication about prices and delivery times

❖ PrintHub aims to eliminate these issues using a simple online system.

Target Users:

❖ **PrintHub is designed for:**

✓ **Students:**

who need printing, photocopying, or lamination regularly.

✓ **Office Staff & Faculty:**

who require quick and professional document printing.

✓ **Small Business Owners**

who print posters, flyers, business cards, and banners.

✓ **Print Shop Operators**

who want a modern system to manage customer orders more efficiently.

Project Goals and Objectives :

The main goals of PrintHub are:

- To provide printing services online without physical waiting
- To display service prices and details clearly
- To allow users to upload files directly from their device
- To generate an automatic queue number for every order
- To track the order status in real time
- To help print shops manage orders smoothly

Key Features In CookBook :

❖ Browse Printing Services:

❖ Users can explore a variety of services such as:

- Xerox (Black & White / Color)
- Document printing
- Photo printing
- ID card printing
- Poster and banner printing
- Lamination and binding
- Business card printing

❖ Each service shows:

- Price
- Delivery time
- Short description
- A “View Details” button

❖ File Upload System

- PDFs
- Images
- Word files
- ID photos

They can preview the uploaded files before submitting.

❖ Online Order Placement

The user selects:

- Service type
- Number of copies
- Paper size
- Print color options
- Single side or double side

Then they confirm the order.

❖ Digital Queue Management

After placing an order:

- A queue number is generated
- The user sees estimated waiting time
- The status updates step-by-step:
 - Pending
 - Processing
 - Completed
 - Ready for Pickup

❖ Admin Dashboard

The print shop admin can:

- View all incoming orders
- Update order status
- Track remaining workloads
- Manage service details and pricing

❖ Real-Time Notifications:

Users receive updates when:

- Order is placed
- Order is being printed
- Order is completed
- Order is ready

❖ Responsive UI Design

The app works smoothly on:

- Mobile
- Desktop
- Tablets

PRE-REQUISITES FOR PRINTHUB:

❖ Technology Stack

Frontend:

- React.js
- Vite
- React Router
- Axios
- Tailwind CSS
- React Icons
- Toast Notifications

❖ Backend / Data Storage

- JSON Server (or MongoDB/Firebase depending on version)

❖ Developer Tools

- Git and GitHub
- VS Code

❖ Project Structure :

```
PrintHub/  
|  
├─ public/  
├─ src/  
|   ├─ components/  
|   ├─ pages/  
|   ├─ context/  
|   └─ App.jsx  
└─ main.jsx  
  
├─ db.json  
├─ index.html  
└─ package.json
```

❖ Component Overview:

ServiceCard Component:

Shows:

- Service icon
- Name
- Price
- Description
- View Details button

❖ **ServiceDetails Page**

Includes:

- Full service info
- Options to choose paper size, copies, etc.
- File upload box
- Confirm order button

❖ **OrderStatus Page**

Shows:

- Queue number
- Estimated waiting time
- Progress bar
- Order details

❖ **Admin Dashboard**

Includes:

- Order list
- Update status buttons
- Customer details
- Service management tools

10. How the System Works (Flow)

- ❑ User opens PrintHub
- ❑ User selects a service
- ❑ User uploads file (if required)
- ❑ User confirms order
- ❑ System generates queue number
- ❑ Admin prints the document
- ❑ Status updates in real time
- ❑ User picks up the printed material

❖ Create a new React project using Vite:

```
npm create vite@latest
```

Enter → type project name → select **React + JavaScript/TypeScript**.

Navigate to project folder:

```
cd project-  
name npm  
install
```

❖ Run the React development server:

```
npm run dev
```

- This starts the **PRINTHUB** app on:
<http://localhost:5173>

- **JSON Server:**

PRINTUB uses a local **JSON Server** to store and manage user-saved favorite recipes.

Start it using:

```
npm run server
```

This runs JSON Server at:

<http://localhost:3001>

❖ HTML, CSS, and JavaScript:

Basic understanding of:

- **HTML** → For structure
- **CSS/Tailwind** → For styling
- **JavaScript** → For logic, API calls, and component interaction

❖ Version Control (Git):

Git is recommended for tracking project changes and collaborating with team members.

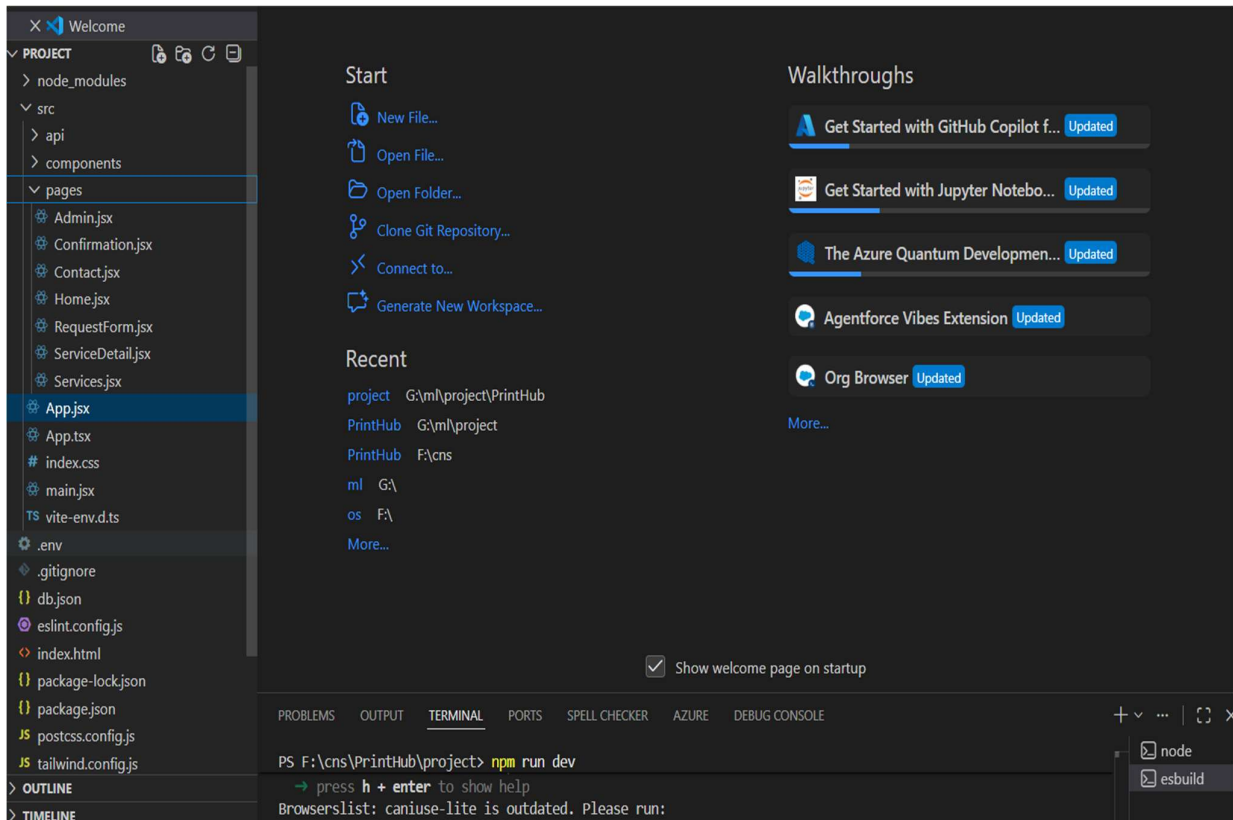
- **Git Download:** <https://git-scm.com/downloads>

• Development Environment:

Use any code editor of your choice:

- **Visual Studio Code:** <https://code.visualstudio.com/download>
- **Sublime Text:** <https://www.sublimetext.com/download>
- **WebStorm:** <https://www.jetbrains.com/webstorm/download>

PROJECT STRUCTURE PRINTHUB:



- ❖ A well-organized folder structure is essential for building scalable applications.
- ❖ Important files from your project:
 - **index.html** → Root HTML for React app
 - **package.json** → Contains major dependencies like axios, react-router-dom, react-icons, JSON server, toastify

package
 - **db.json** → Stores user favorite recipes locally using JSON Server DB

PROJECT FLOW:

- ❑ User opens PrintHub
- ❑ User selects a service
- ❑ User uploads file (if required)
- ❑ User confirms order
- ❑ System generates queue number
- ❑ Admin prints the document
- ❑ Status updates in real time
- ❑ User picks up the printed material

MILESTONE 1: Project Setup & Configuration

1. Install Required Tools:

Open the project folder and install:

- React.js
 - Axios
 - React Router
 - React Icons
-

- React Toastify
- JSON Server

❖ These libraries are visible in your package.json

2. Recommended Documentation:

- <https://react.dev/learn/installation>
- <https://reactrouter.com>
- <https://mui.com>
- <https://www.npmjs.com/package/json-server>

MILESTONE 2: Web Development

1. Setup React Application:

- Create React app
- Configure React Router
- Install required dependencies
- Set up JSON server

2. Design UI Components:

- Develop reusable components:
 - RecipeCard
 - RecipeList
 - Navbar
 - Favorites Page
 - Recipe Details Page
- Implement layout with TailwindCSS
- Add navigation using React Router v6+

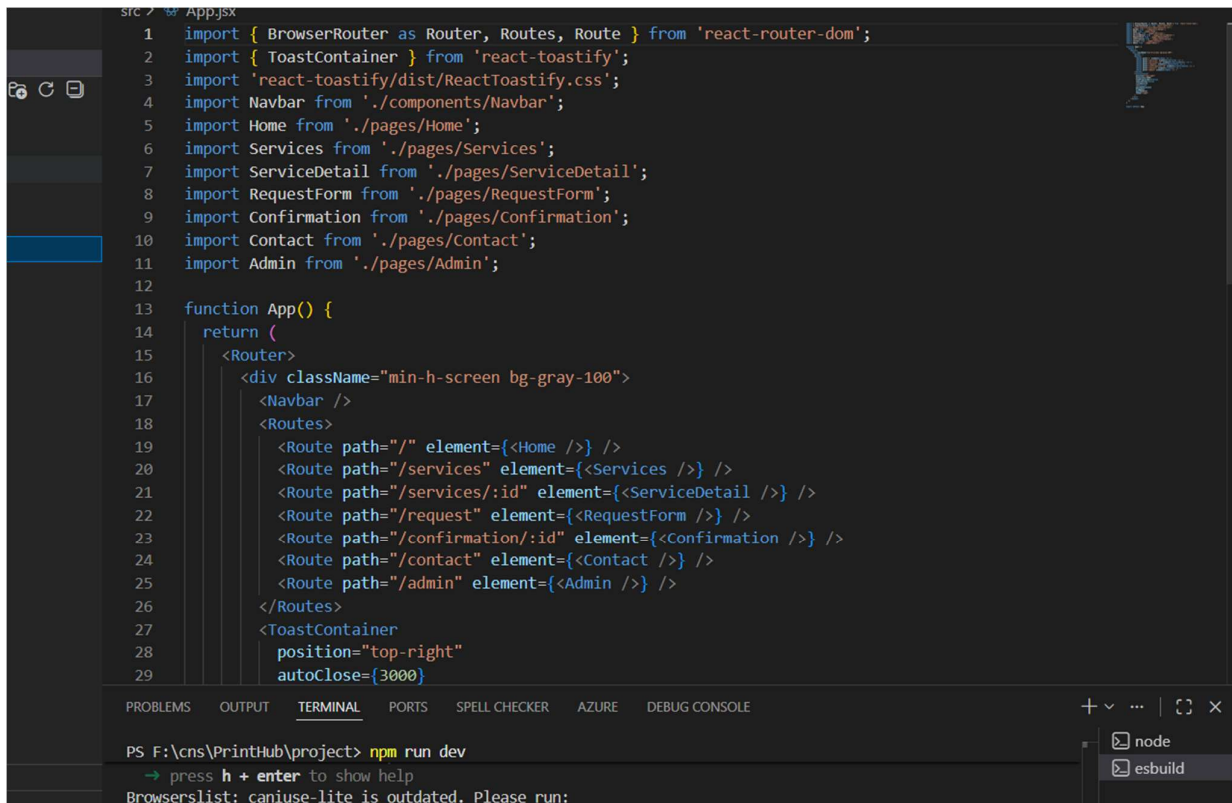
3. Implement Frontend Logic

- Connect to recipe API
- Fetch recipe list dynamically
- Manage favorites using JSON Server
- Bind data into components
- Implement toast notifications
- Add search functionality

DATA PROVIDER FOR PRINTHUB :

Code Description (PRINTHUB Adaptation):

- Creates a RecipeContext using createContext(null) to hold global data such as:
 - recipe list
 - favorites
 - search results
- Defines a RecipeProvider that wraps the app and provides shared state.
- Uses useState or useReducer to track service data.
- Returns a ServiceContext.Provider with values passed to child components.
- Nested components use useContext(Servicetext) to access or update recipes.



```
src / App.jsx
1  import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
2  import { ToastContainer } from 'react-toastify';
3  import 'react-toastify/dist/ReactToastify.css';
4  import Navbar from './components/Navbar';
5  import Home from './pages/Home';
6  import Services from './pages/Services';
7  import ServiceDetail from './pages/ServiceDetail';
8  import RequestForm from './pages/RequestForm';
9  import Confirmation from './pages/Confirmation';
10 import Contact from './pages/Contact';
11 import Admin from './pages/Admin';
12
13 function App() {
14   return (
15     <Router>
16       <div className="min-h-screen bg-gray-100">
17         <Navbar />
18         <Routes>
19           <Route path="/" element={<Home />} />
20           <Route path="/services" element={<Services />} />
21           <Route path="/services/:id" element={<ServiceDetail />} />
22           <Route path="/request" element={<RequestForm />} />
23           <Route path="/confirmation/:id" element={<Confirmation />} />
24           <Route path="/contact" element={<Contact />} />
25           <Route path="/admin" element={<Admin />} />
26         </Routes>
27       </div>
28       <ToastContainer
29         position="top-right"
30         autoClose={3000}
31       />
32     </Router>
33   );
34 }
```

PROBLEMS OUTPUT TERMINAL PORTS SPELL CHECKER AZURE DEBUG CONSOLE

PS F:\cns\PrintHub\project> npm run dev
→ press h + enter to show help
Browserslist: caniuse-lite is outdated. Please run:

node
esbuild

PRINTHUB:

❖ **ServiceCard Component:**

Code Description:

- Imports React, styling tools, and icons.
- Accepts props such as:
 - title, image, time, services, serviceLabels
- Displays recipe details in a card layout.
- Provides a button to “Add to Favorites” using axios → JSON Server endpoint.
- Shows toast feedback for actions.
- Fully responsive with hover animations.

❖ **SERVICES COMPONENT :**

```
src > pages > ServiceDetail.jsx > ServiceDetail / loadService
1 import { useState, useEffect } from 'react';
2 import { useParams, useNavigate, Link } from 'react-router-dom';
3 import { ArrowLeft, Clock, DollarSign, Tag } from 'lucide-react';
4 import { PrimaryButton, SecondaryButton } from '../components/Button';
5 import { servicesApi } from '../api/api';
6 import { toast } from 'react-toastify';
7
8 export default function ServiceDetail() {
9   const { id } = useParams();
10   const navigate = useNavigate();
11   const [service, setService] = useState(null);
12   const [loading, setLoading] = useState(true);
13
14   useEffect(() => {
15     loadService();
16   }, [id]);
17
18   const loadService = async () => {
19     try {
20       setLoading(true);
21       const response = await servicesApi.getById(id);
22       setService(response.data);
23     } catch (error) {}
24     console.error('Error loading service:', error);
25     toast.error('Failed to load service details');
26     navigate('/services');
27   } finally {
28     setLoading(false);
29   }
30 };
31
32 if (loading) {
33   return (
34     <div className="min-h-screen bg-gray-50 flex items-center justify-center">
35       <div className="text-center">
36         <div className="animate-spin rounded-full h-16 w-16 border-b-2 border-[#5E2A84] mx-auto"></div>
37         <p className="mt-4 text-gray-600">Loading service details...</p>

```

```

37     <p className="mt-4 text-gray-600">Loading service details...</p>
38   </div>
39 </div>
40 );
41 }
42
43 if (!service) {
44   return (
45     <div className="min-h-screen bg-gray-50 flex items-center justify-center">
46       <div className="text-center">
47         <p className="text-gray-600">Service not found</p>
48         <Link to="/services" className="text-[#5E2A84] hover:underline mt-4 inline-block">
49           Back to Services
50         </Link>
51       </div>
52     </div>
53   );
54 }
55
56 return (
57   <div className="min-h-screen bg-gray-50 py-12">
58     <div className="max-w-4xl mx-auto px-4 sm:px-6 lg:px-8">
59       <button
60         onClick={() => navigate('/services')}
61         className="flex items-center text-gray-600 hover:text-[#5E2A84] mb-8 transition"
62       >
63         <ArrowLeft className="w-5 h-5 mr-2" />
64         Back to Services
65       </button>
66
67       <div className="bg-white rounded-lg shadow-lg overflow-hidden">
68         <div className="bg-gradient-to-r from-[#5E2A84] to-[#8B4FB3] p-8 text-white">
69           <h1 className="text-4xl font-bold mb-4">{service.name}</h1>
70           {service.popular && (
71             <span className="inline-block bg-white/20 text-white px-4 py-2 rounded-full text-sm for
72               Popular Service

```

```

72     Popular Service
73     </span>
74   })
75 </div>
76
77 <div className="p-8">
78   <div className="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">
79     <div className="flex items-center space-x-3">
80       <div className="bg-purple-100 p-3 rounded-lg">
81         <DollarSign className="w-6 h-6 text-[#5E2A84]" />
82       </div>
83       <div>
84         <p className="text-sm text-gray-600">Price</p>
85         <p className="text-xl font-bold text-gray-800">${service.price}</p>
86       </div>
87     </div>
88
89     <div className="flex items-center space-x-3">
90       <div className="bg-purple-100 p-3 rounded-lg">
91         <Clock className="w-6 h-6 text-[#5E2A84]" />
92       </div>
93       <div>
94         <p className="text-sm text-gray-600">Turnaround</p>
95         <p className="text-xl font-bold text-gray-800">{service.turnaround}</p>
96       </div>
97     </div>
98
99     <div className="flex items-center space-x-3">
100      <div className="bg-purple-100 p-3 rounded-lg">
101        <Tag className="w-6 h-6 text-[#5E2A84]" />
102      </div>
103      <div>
104        <p className="text-sm text-gray-600">Category</p>
105        <p className="text-xl font-bold text-gray-800">{service.category}</p>
106      </div>
107    </div>

```



```

101 |         <tag className="w-8 h-8 text-[#5E2A84]" />
102 |     </div>
103 |     <div>
104 |         <p className="text-sm text-gray-600">Category</p>
105 |         <p className="text-xl font-bold text-gray-800">{service.category}</p>
106 |     </div>
107 | </div>
108 | </div>
109 |
110 | <div className="mb-8">
111 |     <h2 className="text-2xl font-bold text-gray-800 mb-4">Service Description</h2>
112 |     <p className="text-gray-600 leading-relaxed">{service.description}</p>
113 | </div>
114 |
115 | <div className="bg-gray-50 p-6 rounded-lg mb-8">
116 |     <h3 className="text-xl font-bold text-gray-800 mb-4">What's Included</h3>
117 |     <ul className="space-y-2 text-gray-600">
118 |         <li className="flex items-start">
119 |             <span className="text-[#5E2A84] mr-2">•</span>
120 |             <span>Professional quality printing</span>
121 |         </li>
122 |         <li className="flex items-start">
123 |             <span className="text-[#5E2A84] mr-2">•</span>
124 |             <span>Quality control and inspection</span>
125 |         </li>
126 |         <li className="flex items-start">
127 |             <span className="text-[#5E2A84] mr-2">•</span>
128 |             <span>Fast turnaround time</span>
129 |         </li>
130 |         <li className="flex items-start">
131 |             <span className="text-[#5E2A84] mr-2">•</span>
132 |             <span>Customer support throughout the process</span>
133 |         </li>
134 |     </ul>
135 | </div>
136 |

```

0.0.1:13339 Java: Ready Ln 43, Col 18 (523 selected) Spaces: 2 UTF-8 LF JavaScript JSX

```

136 |
137 |     <div className="flex flex-col sm:flex-row gap-4">
138 |         <Link to={` /request?service=${service.id}`} className="flex-1">
139 |             <PrimaryButton className="w-full">Book This Service</PrimaryButton>
140 |         </Link>
141 |         <Link to="/contact" className="flex-1">
142 |             <SecondaryButton className="w-full">Contact Us</SecondaryButton>
143 |         </Link>
144 |     </div>
145 | </div>
146 | </div>
147 | </div>
148 | </div>
149 | );
150 | }
151 |

```

❖ **Code Description:**

❖ **Fetches recipe details from API using service ID.**

❖ **Displays:**

- Normal xerox
- Color Xerox
- Document Printing (B&W)
- Document Printing (Color)
- Multi-Page Printing
- Back-to-Back Printing
- Spiral Binding Printing
- Allows adding to Favorites.

Favorites Component

```

    <p className="text-xl font-bold text-gray-800">{service.turnaround}</p>
  </div>
</div>

<div className="flex items-center space-x-3">
  <div className="bg-purple-100 p-3 rounded-lg">
    <Tag className="w-6 h-6 text-[#5E2A84]" />
  </div>
  <div>
    <p className="text-sm text-gray-600">Category</p>
    <p className="text-xl font-bold text-gray-800">{service.category}</p>
  </div>
</div>
</div>

<div className="mb-8">
  <h2 className="text-2xl font-bold text-gray-800 mb-4">Service Description</h2>
  <p className="text-gray-600 leading-relaxed">{service.description}</p>
</div>

<div className="bg-gray-50 p-6 rounded-lg mb-8">
  <h3 className="text-xl font-bold text-gray-800 mb-4">What's Included</h3>
  <ul className="space-y-2 text-gray-600">
    <li className="flex items-start">
      <span className="text-[#5E2A84] mr-2">•</span>
      <span>Professional quality printing</span>
    </li>
    <li className="flex items-start">
      <span className="text-[#5E2A84] mr-2">•</span>
      <span>Quality control and inspection</span>
    </li>
    <li className="flex items-start">
      <span className="text-[#5E2A84] mr-2">•</span>
      <span>Fast turnaround time</span>
    </li>
  </ul>
</div>

```

Ln 43, Col 18 (523 selected) Spaces: 2 UTF-8 LF { JavaScript JSX

Code Description:

- Loads user-saved recipes from db.json.
- Uses axios to perform CRUD operations.
- Displays saved recipes with option to remove.

Project Execution

After completing the development of the PRINTHUB application, you can run the React project using the command:

```
npm run dev
```

This will start the development server and launch the printhub application in your browser at:

- <http://localhost:5173>

If you are using a JSON Server for storing favorite services, run:

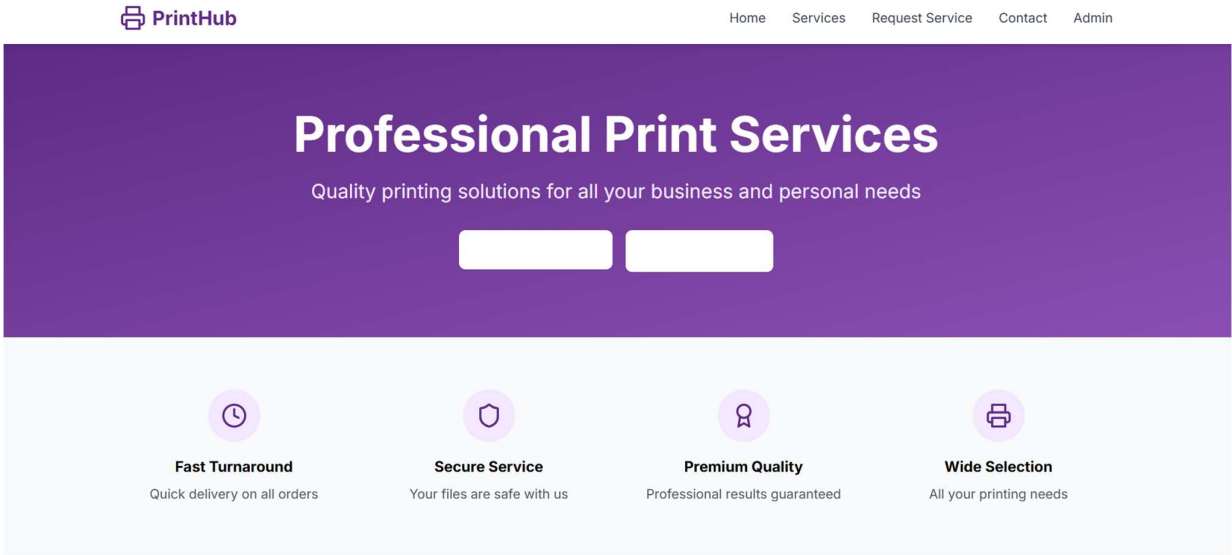
```
npm run server
```

This starts the JSON API at:

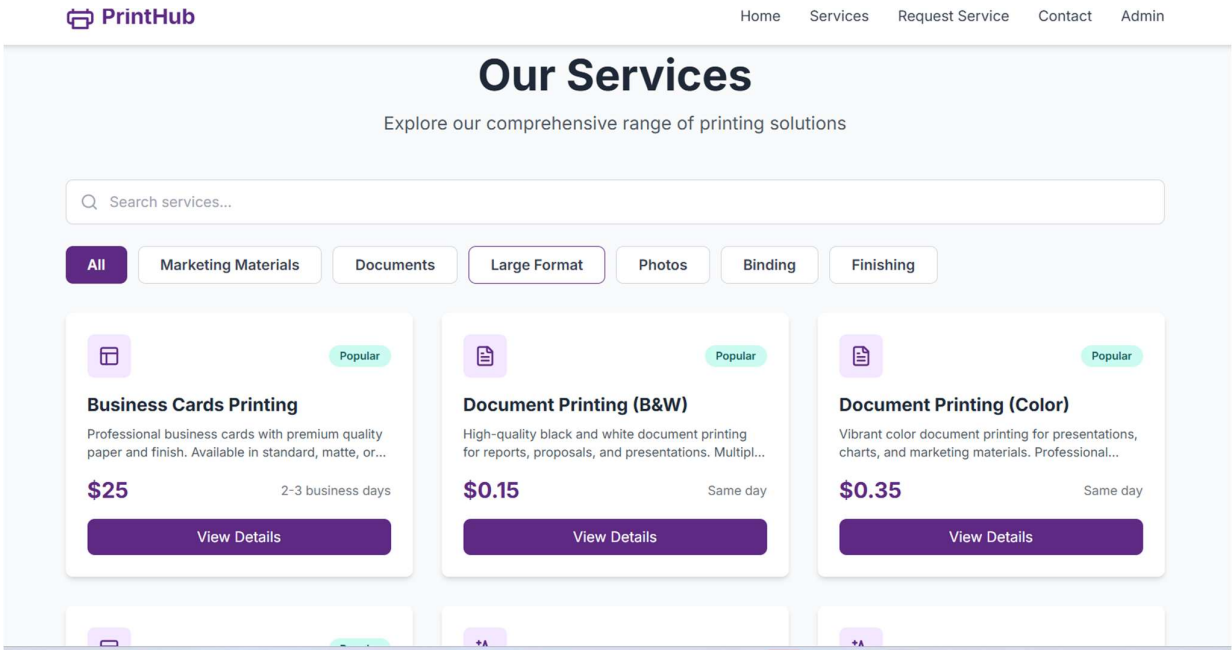
- **<http://localhost:3001>**


Below are some screenshots representing the different components and pages of the application.

● Home Component:



SERVICE LIST COMPONENT:






Popular

Flyer Printing

Eye-catching flyers for events, promotions, and announcements. Various sizes and paper stocks...

\$50 3-5 business days

[View Details](#)




Poster Printing

Large format poster printing for events, advertising, and displays. Available in multiple...

\$15 2-4 business days

[View Details](#)




Banner Printing

Durable vinyl banners for indoor and outdoor use. Custom sizes available with grommets for easy...

\$85 5-7 business days

[View Details](#)



Popular

Brochure Printing

Professional tri-fold or bi-fold brochures. Perfect for marketing materials, product catalogs, and...

\$75 3-5 business days

[View Details](#)





Photo Printing

High-resolution photo prints on premium photo paper. Various sizes from 4×6 to 11×14 inches.

\$2.5 1-2 business days

[View Details](#)




Booklet Binding

Professional booklet binding services including saddle stitch and spiral binding. Ideal for manual...

\$5 3-5 business days

[View Details](#)



Popular

Brochure Printing

Professional tri-fold or bi-fold brochures. Perfect for marketing materials, product catalogs, and...

\$75 3-5 business days

[View Details](#)





Photo Printing

High-resolution photo prints on premium photo paper. Various sizes from 4×6 to 11×14 inches.

\$2.5 1-2 business days

[View Details](#)




Booklet Binding

Professional booklet binding services including saddle stitch and spiral binding. Ideal for manual...

\$5 3-5 business days

[View Details](#)



Lamination Service

Protect your documents with professional lamination. Available in various thicknesses for...

\$3 Same day

[View Details](#)

❖ REQUEST A SERVICE:

Request a Service

Fill out the form below to submit your print request

Full Name *

John Doe

Phone Number *

+1 (555) 123-4567

Email Address *

john@example.com

Select Service *

Select an option

Quantity *

1

Preferred Pickup Time

dd-mm-yyyy --:--

Upload File (Optional)

Choose File

No file chosen

Upload File (Optional)

Choose File

No file chosen

Additional Notes

Any special instructions or requirements...

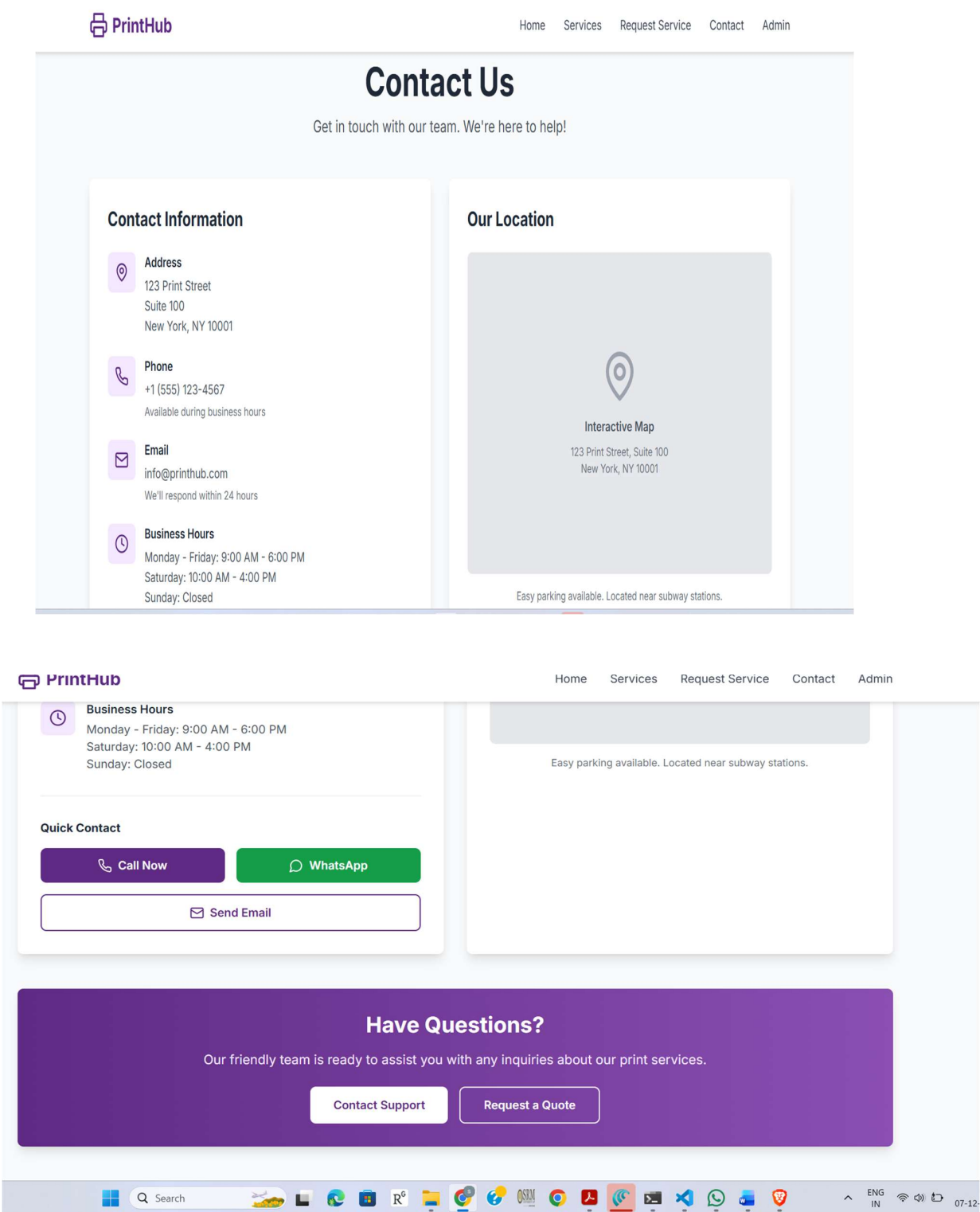
Submit Request

Cancel

What happens next?

1. We'll review your request within 24 hours
2. You'll receive a confirmation email with pricing details
3. Once approved, we'll begin working on your order

❖ CONTACT SUPPORT:



❖ ADMIN DASHBOARD:

Admin Dashboard

Manage services and view customer requests

Services (10)Requests (0)

[+ Add New Service](#)

NAME	CATEGORY	PRICE	TURNAROUND	POPULAR	ACTIONS
Business Cards Printing	Marketing Materials	\$25	2-3 business days	Yes	Edit Delete
Document Printing (B&W)	Documents	\$0.15	Same day	Yes	Edit Delete
Document Printing (Color)	Documents	\$0.35	Same day	Yes	Edit Delete
Flyer Printing	Marketing Materials	\$50	3-5 business days	Yes	Edit Delete
Poster Printing	Large Format	\$15	2-4 business days	No	Edit Delete
Banner Printing	Large Format	\$85	5-7 business days	No	Edit Delete

GITHUB Link

You can view the complete project demonstration at:

Video 1: <https://drive.google.com/file/d/1Xwtq7x2V9PbdbTkAh74vN4PB4KJbyj2X/view?usp=drivesdk>

Video2: <https://drive.google.com/file/d/1XaCpKP1grDHA8OsTQJk3aqY5LTmHiDgT/view?usp=drivesdk>

conclusion:

offers a modern and convenient solution for everyday printing needs. By moving the entire process online PrintHub — from selecting services to uploading files and tracking orders — the system saves users time and reduces crowding in print shops.

It also helps printing operators manage orders more professionally with an organized workflow and clear status updates. The project demonstrates practical use of React.js, user-friendly interface design, and real-world problem solving.

PrintHub is a scalable and useful platform that can be implemented in colleges, offices, and print shops to make printing services faster, simpler, and more efficient.

