▼ FRONTEND SYSTEM DESIGN: URL Shortener

© Goal:

To provide a clean user interface that allows:

- Entering a long URL
- Clicking a "Shorten" button
- Displaying the shortened URL
- Optionally allowing copy/share of the short URL
- Redirecting to long URL when short URL is visited

HIGH-LEVEL ARCHITECTURE

```
[ User (Browser) ]

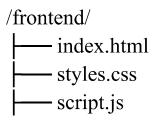
v
[ React / HTML-CSS-JS Frontend ]

v
[ DEST ADJ Colleges Spring Boot Books
```

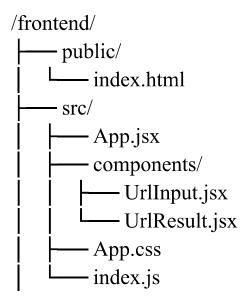
[REST API Calls to Spring Boot Backend]

FOLDER STRUCTURE (Frontend)

If using plain HTML/CSS/JS:



If using React:



UI FLOW

1. Input Page

- Input box to paste the long URL
- Button to generate short URL

2. Output Section

- Shows generated short URL
- "Copy" button (optional)
- Display errors if invalid URL

COMPONENT DESIGN (React)

App.jsx

UrlInput.jsx

```
import React, { useState } from 'react';
import axios from 'axios';
```

```
function UrlInput() {
 const [longUrl, setLongUrl] = useState(");
 const [shortUrl, setShortUrl] = useState(");
 const handleShorten = async () => {
  try {
   const response = await axios.post('http://localhost:8080/shorten', {
longUrl });
   setShortUrl(response.data);
  } catch (err) {
   alert('Error: ' + err.message);
 };
 return (
  <div>
   <input
    type="text"
    placeholder="Enter long URL"
    value={longUrl}
    onChange={(e) => setLongUrl(e.target.value)}
   />
   <button onClick={handleShorten}>Shorten</button>
    {shortUrl && (
    <div>
      Short URL: <a href={shortUrl}>{shortUrl}</a>
    </div>
   )}
  </div>
```

```
);
export default UrlInput;
styles.css (if HTML/CSS/JS)
body {
 font-family: Arial;
 text-align: center;
 margin-top: 80px;
input {
 padding: 10px;
 width: 300px;
 margin-right: 10px;
button {
 padding: 10px 20px;
 cursor: pointer;
```


index.html

<!DOCTYPE html>

```
<html>
<head>
<title>URL Shortener</title>
link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>URL Shortener</h1>
<input type="text" id="longUrl" placeholder="Enter your long URL">
<button onclick="shorten()">Shorten</button>

<script src="script.js"></script>
</body>
</html>

script.js
```

```
async function shorten() {
  const longUrl = document.getElementById("longUrl").value;

const response = await fetch('http://localhost:8080/shorten', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
    },
  body: JSON.stringify({ longUrl })
});

if (response.ok) {
```

```
const shortUrl = await response.text();
document.getElementById("result").innerHTML = `Short URL: <a
href="${shortUrl}">${shortUrl}</a>`;
} else {
alert("Error shortening URL");
}
}
```

STATE FLOW (React Version)

- 1. User types long URL \rightarrow state: longUrl updated
- 2. Click Shorten \rightarrow POST request sent
- 3. Backend responds → state: shortUrl updated
- 4. Short URL displayed with hyperlink

OPTIONAL FEATURES TO ADD LATER

Feature

1 catale	Description
Copy to Clipboard	Add a "Copy" button for short URL
Loading Spinner	Show while waiting for backend

Description

Error Display proper error

Handling messages

URL Regex to validate input

Validation format

URL List Show recent shortened

URLs

If you tell me whether you're using **plain HTML** or **React**, I can customize the exact code accordingly.

Let me know if you also want:

- Deployment steps
- Mobile responsive UI
- Share button / QR code generation