Affordmed URL Shortener Web Application – Design & Implementation Document

> Project Overview

This project is a React-based URL Shortener web app developed as part of the Affordmed Campus Hiring Evaluation. It allows users to shorten URLs, optionally set custom shortcodes and expiry times, and view detailed statistics of link usage—all using client-side storage and logic. The app is responsive and styled with Material UI.

Tech Stack

Layer	Technology	Purpose
Frontend	React (v19)	Component-based SPA
UI Library	Material UI (MUI)	Responsive and styled UI components
Routing	React Router (v7)	Page navigation and redirection
State/Storage	LocalStorage	Client-side URL persistence
Logging	Custom Middleware	Logs actions (SHORTEN, REDIRECT)
Unique IDs	uuid	For generating log keys

Folder & File Structure

java

├— package.json

└─ README.md

CopyEdit affordmed-url-shortener/ ├— public/ | └─ index.html ├— src/ // Custom log middleware using localStorage // Routing setup │ ├— App.jsx // Entry point │ ├— index.js // Basic global styles └── styles.css

★ Core Features

✓ URL Shortening

- Users can input up to 5 URLs at once.
- Optionally specify:
 - Custom shortcode (alphanumeric, unique)
 - o **Expiry time** in minutes (defaults to 30 if not given)
- The short URL is generated in the form:

bash

CopyEdit

http://localhost:3000/abcde

Redirection

- Visiting the short URL via the browser will:
 - o Check expiry.
 - o Log timestamp, referrer source, and location (dummy as "India").
 - o Redirect to the original long URL.

0

Statistics

- Lists all URLs created so far.
- Displays:
 - o Short URL, expiry, creation time
 - o Number of times the URL has been clicked
 - o Click logs: timestamp, source, location

0

K Error Handling

- Invalid URLs are flagged before processing.
- Duplicate shortcodes trigger an error message.
- Expired URLs won't redirect and instead show an alert.

Design Decisions

Area Choice

State Management useState, localStorage for simplicity (no backend required)

Logging Custom logMiddleware() using uuid and localStorage

URL Validation Native URL constructor wrapped in try-catch

Shortcode Generation Random base-36 string (Math.random().toString(36))

Redirection Safety Date comparison for expiry + client-side redirects using window.href

Click Insights Collected using document.referrer and hardcoded location

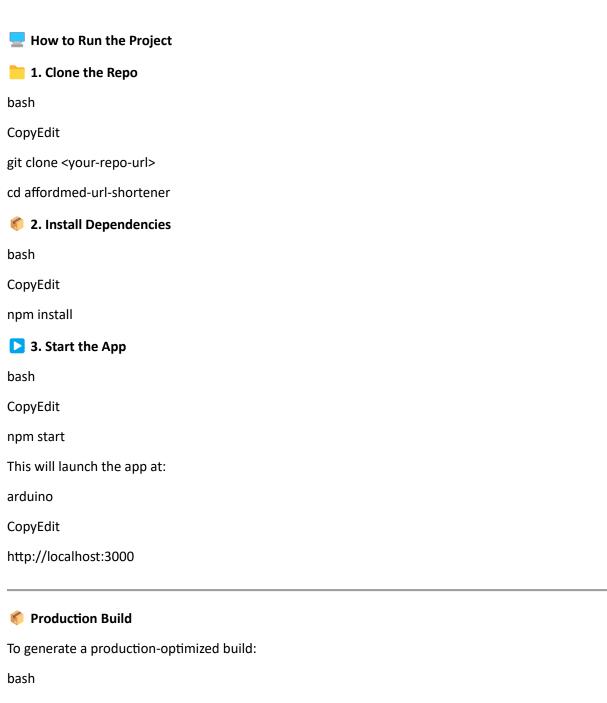
Routing Strategy

Route Pattern Component Purpose

/ HomePage Add and shorten URLs

/stats StatsPage View analytics of all created links

/:shortcode RedirectPage Handle redirect with logging



CopyEdit

npm run build

Assumptions & Constraints

- The application is fully client-side (no authentication or server calls).
- All data persists only in the browser's localStorage.
- Click location is hardcoded due to no backend API for IP geolocation.
- This is an evaluation project and not meant for public deployment.