**🚀 PROJECT TITLE**

**LinkVista – Smart URL Shortener with QR Code & Analytics**

**🎯 PROJECT OBJECTIVE (Problem It Solves)**

In the modern digital world, long URLs are:

* Difficult to share on social media or print media
* Not trackable for click analytics
* Not visually appealing

**LinkVista** solves this by allowing users to:

* Shorten long URLs easily
* Share links via short versions or QR codes
* Track where and how the links are clicked (device, country, time)
* View analytics in a dashboard format

This is **perfect for content creators, businesses, and marketers** who want insights on link engagement.

**💡 UNIQUE SELLING POINT (Compared to Bit.ly, TinyURL)**

* Full dashboard to manage and analyze short links
* Generates QR code for each short link
* Tracks location + device of visitors
* Visual analytics (charts)
* User-friendly, beginner-built, open source

**👥 TARGET USERS**

* Anyone who wants to shorten links (marketers, students, businesses)
* People who want to track engagement (clicks, countries, devices)
* Your potential employers (for your resume and GitHub profile)

**📦 MAIN FEATURES**

**🔐 User Authentication**

* Register with name, email, password
* Login with JWT token
* Show initials as avatar (e.g., "SS" for "Sushmitha Shettigar")

**📌 URL Shortening**

* Paste a long URL
* Generate a short one (like linkvista.vercel.app/xya12b)
* Instant preview of short link and QR code

**📈 Analytics Dashboard**

* See your created links in a table
* Total clicks per link
* Line chart → clicks over time
* Pie chart → devices used (mobile/desktop)
* Pie chart → locations (country-wise)
* Timestamp for each click

**📥 Click Tracking**

When anyone (not just the creator) clicks a short URL:

* Redirects to original URL
* Tracks:
  + Device (mobile or desktop)
  + Location (via IP)
  + Time of click

**📸 QR Code Generation**

* Automatically generate a QR Code for every short link
* Downloadable

**🔄 PROJECT FLOW**

**1. Public Visit**

* Landing Page → Brief intro, Login/Register buttons

**2. Authentication**

* Register (name, email, password)
* Login → Get JWT token → store in localStorage

**3. Dashboard (After Login)**

* Navbar with user initials
* Create short link form
* View table of created links
* Clickable short link
* View QR code
* Visual analytics

**4. Someone Clicks a Short URL**

* Short URL is opened
* Backend:
  + Tracks device, location, time
  + Redirects to original URL

**5. Dashboard Analytics**

* Table view + Chart view of link activity
* Easily track performance of shared links

**🛠 TECH STACK**

| **Layer** | **Tech** |
| --- | --- |
| Frontend | React, React Router, Axios |
| Styling | Plain CSS (.css files) |
| Backend | Node.js + Express.js |
| Database | MongoDB + Mongoose |
| Auth | JWT + bcrypt |
| Charts | Chart.js via react-chartjs-2 |
| QR Code | qrcode.react |
| Location API | ipinfo.io (for IP to country) |
| Device Info | Browser User-Agent |
| Deployment | Netlify (client) + Render (server) |

**📂 FILE STRUCTURE OVERVIEW**

linkvista/

├── client/ # React App

│ ├── src/

│ │ ├── components/ # Navbar, Chart, QR

│ │ ├── pages/ # Home, Login, Register, Dashboard

│ │ ├── App.js

│ │ ├── api.js # Axios config

│ │ └── styles.css # Your custom CSS

├── server/ # Node.js App

│ ├── models/ # User.js, Url.js

│ ├── routes/ # authRoutes, urlRoutes

│ ├── controllers/ # authController, urlController

│ ├── middleware/ # JWT auth

│ └── utils/ # geo + device tracking

**🔑 USER STORIES (UX Perspective)**

1. **As a new user**, I want to register and get a dashboard to create short links.
2. **As a logged-in user**, I want to paste long URLs and get a short one.
3. **As a user**, I want to see QR codes and click stats for each link I created.
4. **As someone clicking a short link**, I should be taken to the original site instantly.
5. **As the creator**, I want to know:
   * How many people clicked
   * From which country
   * Using what device
   * At what time
6. **As a marketer**, I want visual charts to analyze link performance.

**🧠 LEARNING OUTCOMES (Add in Resume or GitHub)**

By building LinkVista, you’ll demonstrate:

* Understanding of the MERN stack
* Secure user auth using JWT
* QR code generation in React
* Fullstack API integration
* Device + IP tracking
* Visualization with chart libraries
* Clean UI using custom CSS (not frameworks)
* Deployment skills (Render + Netlify)
* Writing clean, understandable code from scratch

**🧾 HOW TO PRESENT ON RESUME**

**LinkVista – Fullstack URL Shortener with QR & Analytics**  
Built a secure MERN-based URL shortener that allows users to create, manage, and analyze shortened links.  
Features include login auth, QR code generation, and real-time analytics with device/location tracking.  
Tech: React, Node.js, MongoDB, JWT, Chart.js, CSS  
[Live Demo](https://chatgpt.com/c/68579a67-d524-8010-9a13-1b5e8ba838c4) | [GitHub](https://chatgpt.com/c/68579a67-d524-8010-9a13-1b5e8ba838c4)

Client – npm run dev

Server – npm start

**Install Backend Packages**

npm install express mongoose cors dotenv bcryptjs jsonwebtoken useragent axios

| **Package** | **Purpose** |
| --- | --- |
| express | Build backend server |
| mongoose | Connect and talk to MongoDB |
| cors | Allow frontend to access backend |
| dotenv | Store secrets like DB connection |
| bcryptjs | Encrypt passwords |
| jsonwebtoken | Auth tokens for login |
| useragent | Detect device info (mobile/desktop) |
| axios | (optional) for API requests like IP info |
|  |  |

| **Feature** | **Description** |
| --- | --- |
| 🔐 **User Auth** | Register & Login with JWT |
| ✂️ **URL Shortening** | Generate short URL + store creator info |
| 📎 **QR Code** | Automatically generated as base64 image |
| 📊 **Click Tracking** | Count + device type + geolocation tracking |
| 📂 **Get My URLs** | Authenticated API to list all user's links |
| 🗑️ **Delete URL** | Only creator can delete their own link |
| 📈 **Analytics API** | Stats of clicks by device & location |

**✅ Let’s Quickly Recap the Stack:**

* **Frontend**: React (with normal CSS, not Tailwind)
* **Backend**: Node.js + Express + MongoDB + Mongoose
* **QR Code**: qrcode (Node package)
* **GeoLocation**: geoip-lite
* **Auth**: JWT

**🎯 React Frontend Plan (Step-by-Step)**

We’ll build it feature-by-feature. Here's the UI workflow:

**🔹 Public Landing Page**

* Welcome text
* Login / Register buttons

**🔹 Auth Pages**

* Register page
* Login page

**🔹 Dashboard (after login)**

* NavBar with name initials (A B)
* Show all created URLs with:
  + Short URL
  + QR code
  + Click count
  + Buttons: View Stats 📊 | Delete 🗑️

**🔹 Create URL Page**

* Input full URL
* On submit, show:
  + Short URL
  + QR code

**🔹 Stats Page**

* Bar chart of device type
* Pie chart of locations
* Total click count

**Frontend**

npm install axios react-router-dom chart.js react-chartjs-2

* axios: for API calls
* react-router-dom: for navigation
* chart.js & react-chartjs-2: for charts

src/

├── api/ → Axios config

├── components/ → NavBar, URLCard, Charts

├── pages/ → Login, Register, Dashboard, Stats

├── styles/ → Write plain CSS here

| **Page** | **Purpose** |
| --- | --- |
| **Landing Page** | Public, shows logo, nav, main form (URL input), features, FAQ, footer |
| **Auth Pages** | Login / Signup — used for authentication |
| **Dashboard** | Private (after login), shows stats, charts, links, profile dropdown |

src/

├── api/

│ └── axios.js # Axios config with base URL + auth token

│

├── assets/ # Logo, illustrations

│

├── components/

│ ├── Navbar.jsx

│ ├── Footer.jsx

│ ├── FeatureSection.jsx

│ ├── FAQSection.jsx

│ ├── URLInputBox.jsx # Main shorten form (used in LandingPage)

│ ├── URLCard.jsx # Shows individual shortened links in dashboard

│ ├── ChartSection.jsx # Device/location charts

│ └── ProfileDropdown.jsx # Dropdown after login

│

├── context/

│ └── AuthContext.js # Global user state management

│

├── pages/

│ ├── LandingPage.jsx

│ ├── Login.jsx

│ ├── Register.jsx

│ ├── Dashboard.jsx

│ └── NotFound.jsx

│

├── styles/

│ ├── navbar.css

│ ├── landing.css

│ ├── auth.css

│ ├── dashboard.css

│ └── common.css

│

├── App.jsx

└── main.jsx

| **Feature** | **Tool** |
| --- | --- |
| Routing | react-router-dom |
| API Calls | axios |
| Auth Token | Context API + LocalStorage |
| Charts | chart.js + react-chartjs-2 |
| CSS | ✅ Normal CSS files (no Tailwind or ShadCN) |

 AuthContext setup ✅

 Routing setup in App.jsx ✅

 Build Navbar with logic to show login/signup or user initials ✅

 Landing page with sections ✅

 Auth pages (with redirects & messages) ✅

 Dashboard layout ✅

 API integration ✅

 Responsive charts and graphs ✅

**✅ Step-by-Step Plan: Landing Page**

We'll break it down cleanly:

**🎯 Sections on Landing Page:**

1. **Navbar**
   * Left: Logo (text or image)
   * Right: Login & Signup buttons (before login)
   * After login: Show user initials (dropdown with "Dashboard" and "Logout")
2. **Main Hero Section**
   * Big heading (title)
   * Input field to enter long URL
   * Button to "Generate Short Link"
   * ➡️ If not logged in and user tries to submit, redirect to login with message "Please login to generate links"
3. **Feature Section** (3–4 bullet points)
4. **FAQ Section** (accordion-style optional, or simple Q&A format)
5. **Footer** (copyright, credits, etc.)

**✅ Let's Do First: Basic Landing Page Layout**

**🧾 Files We Will Touch Now:**

* src/pages/LandingPage.jsx
* src/components/Navbar.jsx
* src/components/Footer.jsx
* src/components/URLInputBox.jsx
* src/styles/landing.css
* src/App.jsx

{

icon: <FaLink size={28} />,

title: "Shorten Links",

desc: "Turn long URLs into short, easy-to-share links.",

},

{

icon: <FaQrcode size={28} />,

title: "QR Code Ready",

desc: "Auto-generate downloadable QR codes for every short link.",

},

{

icon: <FaChartBar size={28} />,

title: "Track Clicks",

desc: "See who clicked your links, when, and from where.",

},

{

icon: <FaShieldAlt size={28} />,

title: "Safe & Secure",

desc: "User authentication and secure link redirection built-in.",

},

{

q: "How do I create a short link?",

a: "Paste your long URL in the input box, and click 'Generate Short Link'.",

},

{

q: "Do I need to register?",

a: "Yes, registration is needed to manage your links and view analytics.",

},

{

q: "What data does LinkVista track?",

a: "It tracks device type, country, and time of each click — no personal data.",

},

{

q: "Is it free to use?",

a: "Yes, it's completely free and open source.",

},

{

q: "Can I see analytics for my links?",

a: "Yes, your dashboard shows charts, QR codes, and click stats for each link.",

},