



4. URL Shortener

✓ URL Schema Design

Field	Type	Required	Description
<code>_id</code>	<code>ObjectId</code>	Yes	Auto-generated unique ID
<code>originalUrl</code>	<code>String</code>	Yes	The long URL to redirect to
<code>shortCode</code>	<code>String</code>	Yes	Unique code used in the shortened URL
<code>shortUrl</code>	<code>String</code>	Yes	Full shortened URL with domain
<code>clicks</code>	<code>Number</code>	No	How many times the link was visited
<code>createdAt</code>	<code>Date</code>	No	Timestamp when the short URL was created
<code>expiresAt</code>	<code>Date</code>	No	(Optional) When the short URL should expire

✓ URL Shortener API Endpoints

Method	Route	Description
<code>POST</code>	<code>/api/shorten</code>	Create a shortened URL
<code>GET</code>	<code>/:shortCode</code>	Redirect to original URL using shortCode

✓ Core Dependencies (with `nodemon` + `express-rate-limit`)

Package	Purpose
<code>express</code>	Web framework for APIs
<code>mongoose</code>	MongoDB object modeling
<code>nanoid</code>	Generate unique short codes

<code>cors</code>	Enable cross-origin requests
<code>dotenv</code>	Manage environment variables (<code>.env</code>)
<code>morgan</code>	API logging (useful in development)
<code>nodemon</code>	Dev tool for auto-restarting server on file changes
<code>express-rate-limit</code>	Limit repeated requests to public APIs (basic security)