**Ex:** 1

**Date:** 25.6.25

**Development of Toy Store using NodeJS Server without Express**

**AIM:**

To develop an toy store shop website and host it using a NodeJS static server created with core modules such as http, fs and path without using any external frameworks like Express, to serve static HTML ,CSS ,and JavaScript files to the client brower.

**PROCEDURE:**

**STEP 1:Project Initialization:** Create a new project folder (e.g., static-server).

**STEP 2;Navigate to Folder:** Open a terminal or command prompt and navigate into the newly created project folder.

**STEP3:Create package.json:** Run the command npm init -y to create a default package.json file.

**STEP 4:Create HTML File:** Create a new file named index.html in the project root.

**STEP 5:Create CSS File:** Create a second file named style.css in the project root.

**STEP 6:Create Server File:** Create the main server logic file named server.js.

**STEP 7:Write HTML Content:** Add the necessary HTML boilerplate to index.html, including a heading and a <link> tag pointing to /style.css.

**STEP 8:Write CSS Rules:** Add CSS rules to style.css to style the elements from the HTML file.

**STEP 9:Import Modules:** In server.js, import the required Node.js modules: http, fs, and path.

**STEP 10:Implement Server Logic:** Write the code to create an HTTP server that listens for requests.

**STEP 11:Implement Routing:** Inside the server, check the request URL (req.url). If it's '/', serve index.html. If it's '/style.css', serve style.css.

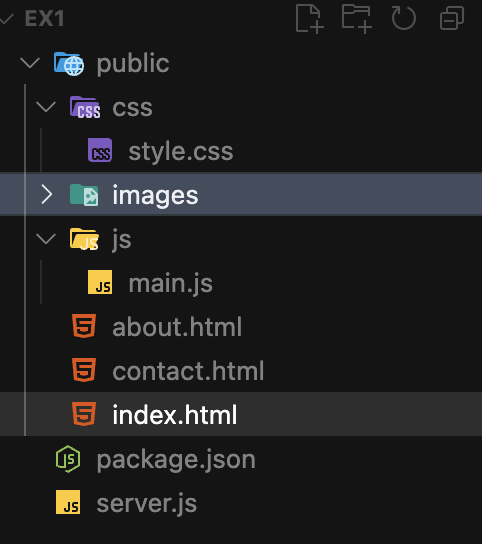
**STEP 12:Set Content-Type:** For each file type, set the correct MIME type in the response header (text/html for HTML, text/css for CSS).

**STEP 13:Handle Errors:** Implement a basic 404 "Not Found" response for any other requested URL.

**STEP 14:Start the Server:** Add the code to make the server listen on a specific port (e.g., 3000).

**STEP 15:Run and Test:** Execute the server using the command node server.js and open http://localhost:3000 in a web browser to verify the result.

**DESIGN:**



1.You start the application by running the command node server.js in your terminal from the EX1 directory.

2.The server.js script executes, creating a web server that starts listening for requests on port 3000.

3.A user opens their web browser and navigates to the URL http://localhost:3000/.

4.The Node.js server receives a request for the root path (/).

5.The server's logic is programmed to map this root request to the index.html file.

6.The server reads the index.html file from its directory and sends its content back to the browser as a response.

7.The browser receives the HTML and begins to render it. While parsing the HTML, it discovers tags linking to /style.css and /script.js.

8.The browser then sends two new, separate requests to the server: one for /style.css and one for /script.js.

9.The server handles these requests by finding the corresponding files in its directory and sending their content back.

10.The browser receives the CSS and applies it to the HTML, then receives and executes the JavaScript. The final, styled, and interactive webpage is then displayed to the user.

**IMPLEMENTATION:**

**1)INDEX.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Toy Galaxy - Your Favorite Toy Store</title>

<link rel="stylesheet" href="/css/style.css">

<link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">

</head>

<body>

<header class="header">

<nav class="nav">

<div class="nav-brand">

<img src="/images/logo.png" alt="Toy Galaxy Logo" width="40" height="40" style="vertical-align: middle; margin-right: 10px;">

<h1>🧸 Toy Galaxy</h1>

</div>

<ul class="nav-menu">

<li><a href="/" class="nav-link active">Home</a></li>

<li><a href="/about.html" class="nav-link">About</a></li>

<li><a href="/contact.html" class="nav-link">Contact</a></li>

</ul>

</nav>

</header>

<main class="main">

<section class="hero">

<div class="container">

<div class="hero-content">

<h1 class="hero-title">Welcome to Toy Galaxy!</h1>

<p class="hero-subtitle">

Discover a universe of fun! Toy Galaxy offers the best selection of toys for kids of all ages. From classic favorites to the latest trends, we have something for every child's imagination.

</p>

<div class="hero-buttons">

<a href="/about.html" class="btn btn-primary">Explore Our Story</a>

<a href="/contact.html" class="btn btn-secondary">Visit Our Store</a>

</div>

</div>

</div>

</section>

<section class="features">

<div class="container">

<h2 class="section-title">Why Shop at Toy Galaxy?</h2>

<div class="features-grid">

<div class="feature-card">

<div class="feature-icon">🧸</div>

<h3>Wide Selection</h3>

<p>From plushies to puzzles, action figures to art kits, we have toys for every interest and age group.</p>

</div>

<div class="feature-card">

<div class="feature-icon">🚚</div>

<h3>Fast Delivery</h3>

<p>Get your favorite toys delivered to your doorstep quickly and safely.</p>

</div>

<div class="feature-card">

<div class="feature-icon">🎁</div>

<h3>Gift Wrapping</h3>

<p>Make every occasion special with our beautiful gift wrapping options.</p>

</div>

<div class="feature-card">

<div class="feature-icon">⭐</div>

<h3>Customer Satisfaction</h3>

<p>We're dedicated to making every shopping experience joyful and memorable.</p>

</div>

</div>

</div>

</section>

<section class="cta">

<div class="container">

<div class="cta-content">

<h2>Ready for Playtime?</h2>

<p>Visit Toy Galaxy today and find the perfect toy for your little star!</p>

<a href="/contact.html" class="btn btn-primary">Contact Us</a>

</div>

</div>

</section>

</main>

<footer class="footer">

<div class="container">

<p>&copy; 2024 Toy Galaxy. Bringing smiles to kids everywhere!</p>

</div>

</footer>

<script src="/js/main.js"></script>

</body>

</html>

**2)STYLE.CSS**

*/\* Reset and Base Styles \*/*

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

html {

scroll-behavior: smooth;

}

body {

font-family: 'Inter', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;

line-height: 1.6;

color: #333;

background-color: #fafafa;

}

*/\* Container \*/*

.container {

max-width: 1200px;

margin: 0 auto;

padding: 0 20px;

}

*/\* Header and Navigation \*/*

.header {

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

color: white;

padding: 1rem 0;

position: sticky;

top: 0;

z-index: 1000;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

}

.nav {

display: flex;

justify-content: space-between;

align-items: center;

}

.nav-brand h1 {

font-size: 1.5rem;

font-weight: 700;

}

.nav-menu {

display: flex;

list-style: none;

gap: 2rem;

}

.nav-link {

color: white;

text-decoration: none;

font-weight: 500;

transition: opacity 0.3s ease;

position: relative;

}

.nav-link:hover,

.nav-link.active {

opacity: 0.8;

}

.nav-link.active::after {

content: '';

position: absolute;

bottom: -5px;

left: 0;

width: 100%;

height: 2px;

background-color: white;

border-radius: 1px;

}

*/\* Main Content \*/*

.main {

min-height: calc(100vh - 140px);

}

*/\* Hero Section \*/*

.hero {

padding: 4rem 0;

background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);

}

.hero .container {

display: grid;

grid-template-columns: 1fr 1fr;

gap: 4rem;

align-items: center;

}

.hero-title {

font-size: 3rem;

font-weight: 700;

margin-bottom: 1rem;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

background-clip: text;

}

.hero-subtitle {

font-size: 1.2rem;

color: #666;

margin-bottom: 2rem;

line-height: 1.8;

}

.hero-buttons {

display: flex;

gap: 1rem;

flex-wrap: wrap;

}

*/\* Code Block \*/*

.code-block {

background: #1e1e1e;

border-radius: 12px;

overflow: hidden;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);

}

.code-header {

background: #2d2d2d;

padding: 1rem;

display: flex;

gap: 0.5rem;

}

.code-dot {

width: 12px;

height: 12px;

border-radius: 50%;

}

.code-dot.red { background: #ff5f56; }

.code-dot.yellow { background: #ffbd2e; }

.code-dot.green { background: #27ca3f; }

.code-content {

padding: 1.5rem;

margin: 0;

color: #d4d4d4;

font-family: 'Monaco', 'Menlo', 'Ubuntu Mono', monospace;

font-size: 0.9rem;

line-height: 1.5;

overflow-x: auto;

}

.code-content code {

color: inherit;

}

*/\* Buttons \*/*

.btn {

display: inline-block;

padding: 0.75rem 1.5rem;

border-radius: 8px;

text-decoration: none;

font-weight: 500;

transition: all 0.3s ease;

border: none;

cursor: pointer;

font-size: 1rem;

}

.btn-primary {

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

color: white;

}

.btn-primary:hover {

transform: translateY(-2px);

box-shadow: 0 10px 20px rgba(102, 126, 234, 0.3);

}

.btn-secondary {

background: white;

color: #667eea;

border: 2px solid #667eea;

}

.btn-secondary:hover {

background: #667eea;

color: white;

transform: translateY(-2px);

}

*/\* Features Section \*/*

.features {

padding: 4rem 0;

background: white;

}

.section-title {

text-align: center;

font-size: 2.5rem;

font-weight: 700;

margin-bottom: 3rem;

color: #333;

}

.features-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 2rem;

}

.feature-card {

background: white;

padding: 2rem;

border-radius: 12px;

text-align: center;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.feature-card:hover {

transform: translateY(-5px);

box-shadow: 0 15px 30px rgba(0, 0, 0, 0.12);

}

.feature-icon {

font-size: 3rem;

margin-bottom: 1rem;

}

.feature-card h3 {

font-size: 1.3rem;

font-weight: 600;

margin-bottom: 1rem;

color: #333;

}

.feature-card p {

color: #666;

line-height: 1.6;

}

*/\* CTA Section \*/*

.cta {

padding: 4rem 0;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

color: white;

text-align: center;

}

.cta-content h2 {

font-size: 2.5rem;

font-weight: 700;

margin-bottom: 1rem;

}

.cta-content p {

font-size: 1.2rem;

margin-bottom: 2rem;

opacity: 0.9;

}

*/\* Page Header \*/*

.page-header {

padding: 3rem 0;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

color: white;

text-align: center;

}

.page-title {

font-size: 2.5rem;

font-weight: 700;

margin-bottom: 1rem;

}

.page-subtitle {

font-size: 1.2rem;

opacity: 0.9;

}

*/\* Content Grid \*/*

.content-grid {

display: grid;

grid-template-columns: 2fr 1fr;

gap: 3rem;

padding: 3rem 0;

}

.content-main h2 {

font-size: 2rem;

font-weight: 600;

margin-bottom: 1rem;

color: #333;

}

.content-main h3 {

font-size: 1.5rem;

font-weight: 600;

margin: 2rem 0 1rem 0;

color: #333;

}

.content-main p {

margin-bottom: 1rem;

color: #666;

line-height: 1.7;

}

.content-main ul {

margin: 1rem 0;

padding-left: 2rem;

}

.content-main li {

margin-bottom: 0.5rem;

color: #666;

}

.content-main code {

background: #f1f3f4;

padding: 0.2rem 0.4rem;

border-radius: 4px;

font-family: 'Monaco', 'Menlo', 'Ubuntu Mono', monospace;

font-size: 0.9rem;

}

.content-main pre {

background: #1e1e1e;

color: #d4d4d4;

padding: 1.5rem;

border-radius: 8px;

overflow-x: auto;

margin: 1rem 0;

font-family: 'Monaco', 'Menlo', 'Ubuntu Mono', monospace;

font-size: 0.9rem;

line-height: 1.5;

}

*/\* Sidebar \*/*

.sidebar-card {

background: white;

padding: 1.5rem;

border-radius: 12px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);

margin-bottom: 1.5rem;

}

.sidebar-card h3 {

font-size: 1.2rem;

font-weight: 600;

margin-bottom: 1rem;

color: #333;

}

.sidebar-card ol,

.sidebar-card ul {

padding-left: 1.5rem;

}

.sidebar-card li {

margin-bottom: 0.5rem;

color: #666;

}

.sidebar-card code {

background: #f1f3f4;

padding: 0.2rem 0.4rem;

border-radius: 4px;

font-family: 'Monaco', 'Menlo', 'Ubuntu Mono', monospace;

font-size: 0.9rem;

}

*/\* Contact Section \*/*

.contact {

padding: 3rem 0;

}

.contact-grid {

display: grid;

grid-template-columns: 1fr 1fr;

gap: 3rem;

}

.contact-info h2 {

font-size: 2rem;

font-weight: 600;

margin-bottom: 1rem;

color: #333;

}

.contact-info > p {

color: #666;

margin-bottom: 2rem;

line-height: 1.7;

}

.contact-item {

display: flex;

align-items: center;

gap: 1rem;

margin-bottom: 1.5rem;

}

.contact-icon {

font-size: 1.5rem;

width: 50px;

height: 50px;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

border-radius: 50%;

display: flex;

align-items: center;

justify-content: center;

color: white;

}

.contact-item h3 {

font-size: 1.1rem;

font-weight: 600;

margin-bottom: 0.25rem;

color: #333;

}

.contact-item p {

color: #666;

}

*/\* Contact Form \*/*

.contact-form h2 {

font-size: 2rem;

font-weight: 600;

margin-bottom: 1.5rem;

color: #333;

}

.form {

background: white;

padding: 2rem;

border-radius: 12px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);

}

.form-group {

margin-bottom: 1.5rem;

}

.form-group label {

display: block;

margin-bottom: 0.5rem;

font-weight: 500;

color: #333;

}

.form-group input,

.form-group textarea {

width: 100%;

padding: 0.75rem;

border: 2px solid #e1e5e9;

border-radius: 8px;

font-size: 1rem;

transition: border-color 0.3s ease;

}

.form-group input:focus,

.form-group textarea:focus {

outline: none;

border-color: #667eea;

}

*/\* FAQ Section \*/*

.faq {

padding: 3rem 0;

background: #f8f9fa;

}

.faq-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 2rem;

}

.faq-item {

background: white;

padding: 2rem;

border-radius: 12px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);

}

.faq-item h3 {

font-size: 1.2rem;

font-weight: 600;

margin-bottom: 1rem;

color: #333;

}

.faq-item p {

color: #666;

line-height: 1.6;

}

.faq-item code {

background: #f1f3f4;

padding: 0.2rem 0.4rem;

border-radius: 4px;

font-family: 'Monaco', 'Menlo', 'Ubuntu Mono', monospace;

font-size: 0.9rem;

}

*/\* Footer \*/*

.footer {

background: #333;

color: white;

text-align: center;

padding: 2rem 0;

}

.footer p {

opacity: 0.8;

}

*/\* Error Pages \*/*

.error-container {

text-align: center;

padding: 4rem 2rem;

min-height: 60vh;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

.error-container h1 {

font-size: 6rem;

font-weight: 700;

color: #667eea;

margin-bottom: 1rem;

}

.error-container h2 {

font-size: 2rem;

font-weight: 600;

margin-bottom: 1rem;

color: #333;

}

.error-container p {

font-size: 1.2rem;

color: #666;

margin-bottom: 2rem;

}

*/\* Responsive Design \*/*

@media (max-width: 768px) {

.nav {

flex-direction: column;

gap: 1rem;

}

.nav-menu {

gap: 1rem;

}

.hero .container {

grid-template-columns: 1fr;

gap: 2rem;

text-align: center;

}

.hero-title {

font-size: 2.5rem;

}

.hero-buttons {

justify-content: center;

}

.content-grid {

grid-template-columns: 1fr;

gap: 2rem;

}

.contact-grid {

grid-template-columns: 1fr;

gap: 2rem;

}

.features-grid {

grid-template-columns: 1fr;

}

.faq-grid {

grid-template-columns: 1fr;

}

.section-title {

font-size: 2rem;

}

.page-title {

font-size: 2rem;

}

.error-container h1 {

font-size: 4rem;

}

.error-container h2 {

font-size: 1.5rem;

}

}

@media (max-width: 480px) {

.container {

padding: 0 15px;

}

.hero-title {

font-size: 2rem;

}

.hero-subtitle {

font-size: 1rem;

}

.btn {

padding: 0.6rem 1.2rem;

font-size: 0.9rem;

}

.nav-menu {

flex-direction: column;

gap: 0.5rem;

}

}

**3)server.js**

const http = require('http');

const fs = require('fs');

const path = require('path');

const url = require('url');

const PORT = process.env.PORT || 3000;

*// MIME types mapping*

const mimeTypes = {

'.html': 'text/html',

'.css': 'text/css',

'.js': 'text/javascript',

'.json': 'application/json',

'.png': 'image/png',

'.jpg': 'image/jpeg',

'.jpeg': 'image/jpeg',

'.gif': 'image/gif',

'.svg': 'image/svg+xml',

'.ico': 'image/x-icon',

'.txt': 'text/plain'

};

*// Function to get MIME type based on file extension*

function **getMimeType**(*filePath*) {

const extname = path.extname(*filePath*).toLowerCase();

return mimeTypes[extname] || 'application/octet-stream';

}

*// Function to serve static files*

function **serveStaticFile**(*filePath*, *res*) {

fs.readFile(*filePath*, (*err*, *data*) => {

if (*err*) {

if (*err*.code === 'ENOENT') {

*// File not found*

*res*.writeHead(404, { 'Content-Type': 'text/html' });

*res*.end(`

<!DOCTYPE html>

<html>

<head>

<title>404 - Page Not Found</title>

<link rel="stylesheet" href="/css/style.css">

</head>

<body>

<div class="error-container">

<h1>404</h1>

<h2>Page Not Found</h2>

<p>The page you're looking for doesn't exist.</p>

<a href="/" class="btn">Go Home</a>

</div>

</body>

</html>

`);

} else {

*// Server error*

*res*.writeHead(500, { 'Content-Type': 'text/html' });

*res*.end(`

<!DOCTYPE html>

<html>

<head>

<title>500 - Server Error</title>

<link rel="stylesheet" href="/css/style.css">

</head>

<body>

<div class="error-container">

<h1>500</h1>

<h2>Server Error</h2>

<p>Something went wrong on our end.</p>

<a href="/" class="btn">Go Home</a>

</div>

</body>

</html>

`);

}

return;

}

*// Set appropriate headers*

const mimeType = getMimeType(filePath);

res.writeHead(200, { 'Content-Type': mimeType });

res.end(data);

});

}

*// Create HTTP server*

const server = http.createServer((*req*, *res*) => {

*// Parse the URL*

const parsedUrl = url.parse(req.url);

let pathname = parsedUrl.pathname;

*// Handle root path*

if (pathname === '/') {

pathname = '/index.html';

}

*// Security: Prevent directory traversal*

if (pathname.includes('..')) {

res.writeHead(403, { 'Content-Type': 'text/html' });

res.end('<h1>403 Forbidden</h1>');

return;

}

*// Construct file path*

const filePath = path.join(\_\_dirname, 'public', pathname);

*// Check if file exists*

fs.access(filePath, fs.constants.F\_OK, (*err*) => {

if (err) {

*// File doesn't exist, serve 404*

res.writeHead(404, { 'Content-Type': 'text/html' });

res.end(`

<!DOCTYPE html>

<html>

<head>

<title>404 - Page Not Found</title>

<link rel="stylesheet" href="/css/style.css">

</head>

<body>

<div class="error-container">

<h1>404</h1>

<h2>Page Not Found</h2>

<p>The page you're looking for doesn't exist.</p>

<a href="/" class="btn">Go Home</a>

</div>

</body>

</html>

`);

} else {

*// File exists, serve it*

serveStaticFile(filePath, res);

}

});

});

*// Start the server*

server.listen(PORT, () => {

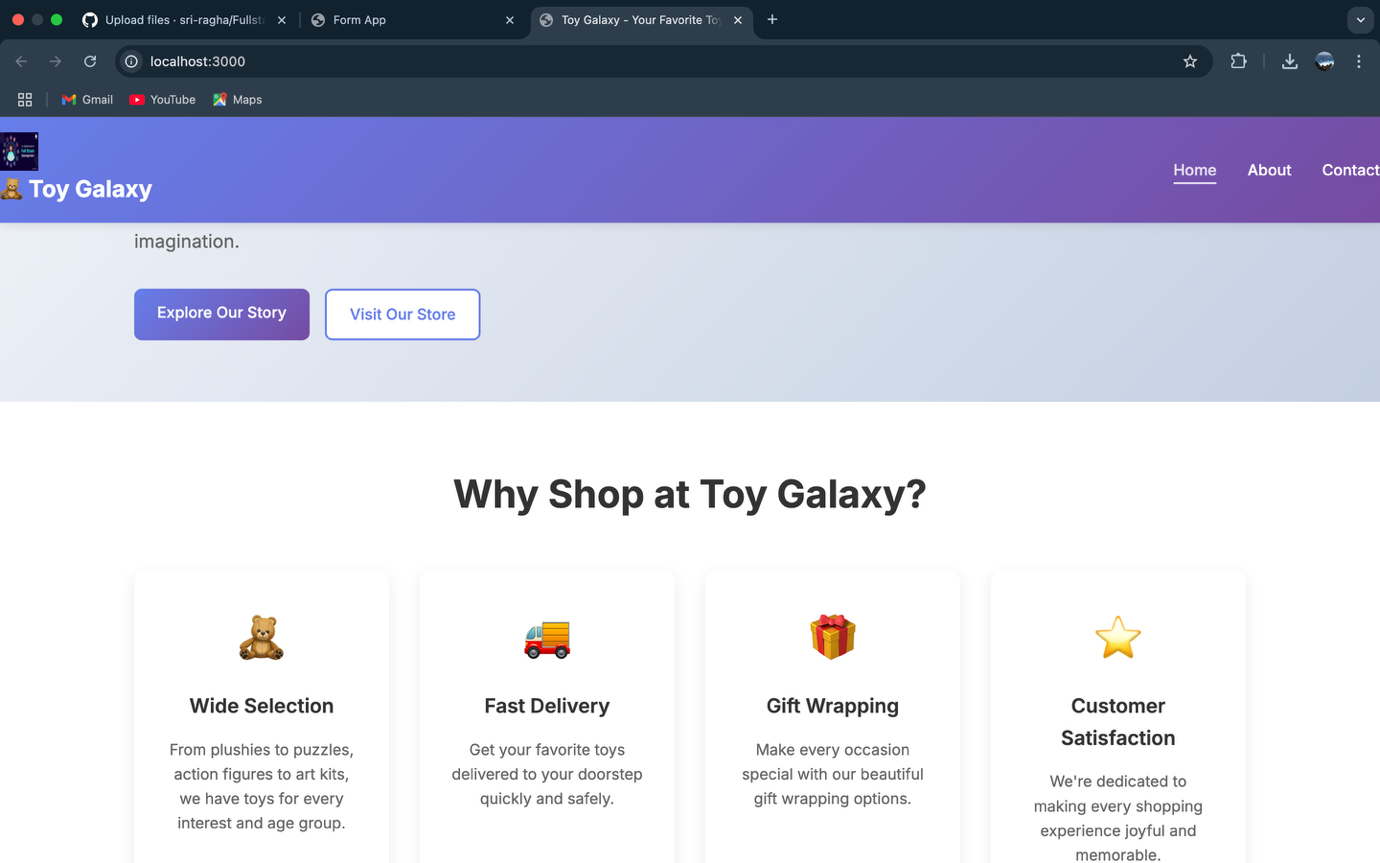
console.log(`🚀 Server running at http://localhost:${PORT}`);

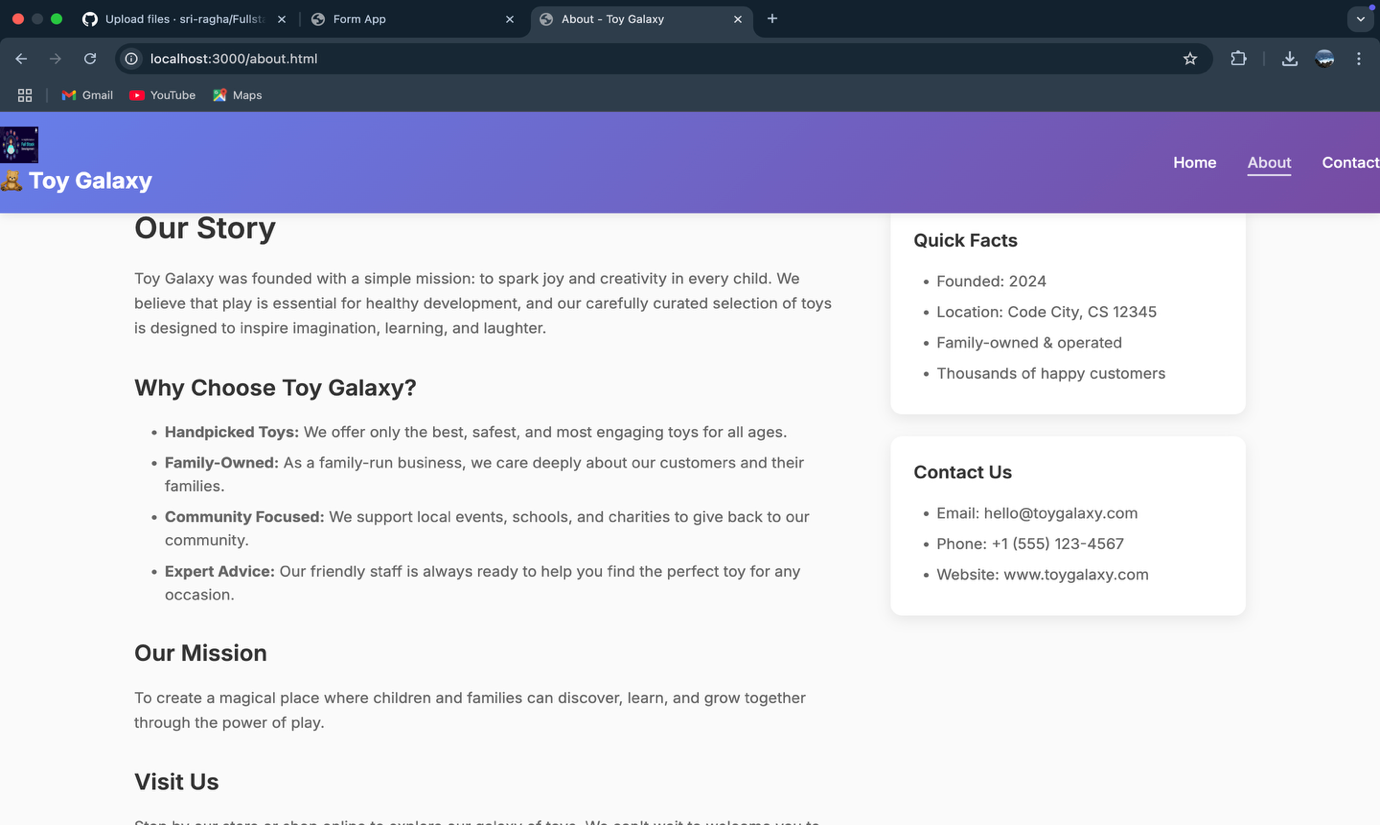
console.log(`📁 Serving static files from ./public directory`);

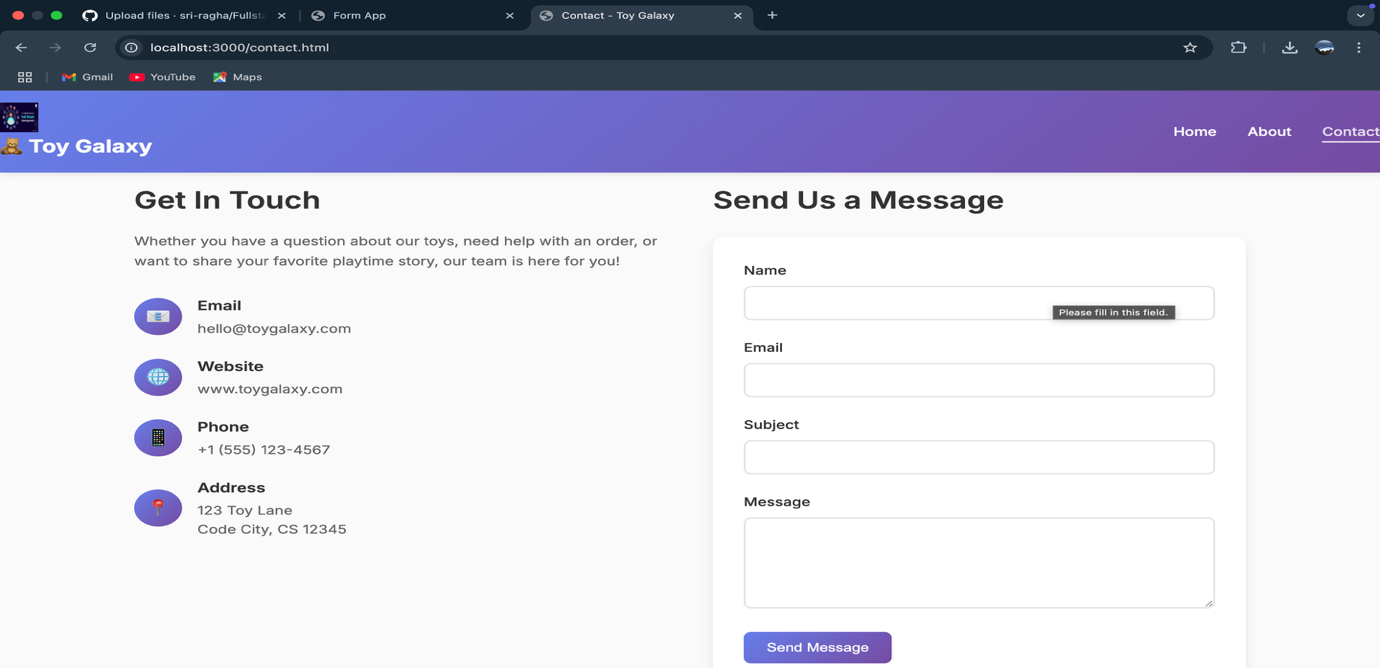
console.log(`🛑 Press Ctrl+C to stop the server`);

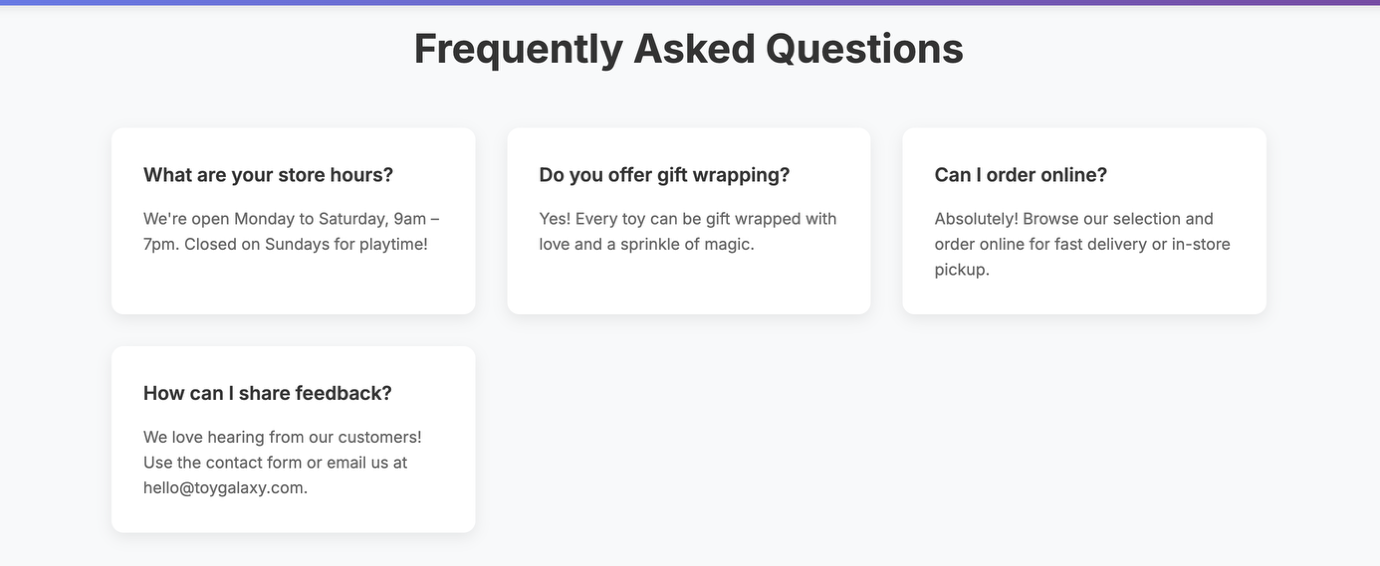
});

**OUTPUT SCREEN SHOT:**

****

****

****

****

|  |  |
| --- | --- |
| **COE(30)** |  |
| **RECORD(20)** |  |
| **VIVA(10)** |  |
| **TOTAL(60)** |  |

**RESULT:**

The online Toy store website was successfully hosted using a NodeJS static server without Express. The static file were served correctly and the application worked as expected at <http://localhost:3000/>.