🛍️ Appscrip Frontend Assignment – Bonaboina Chithra Gowtham

Live URL: https://appscrit-task-gowtham.netlify.app/
GitHub Repository: https://github.com/Tinku070/Appscrip-task-Bonaboina_Chithra_Gowtham.git

---

📘 1. Objective

The goal of this project is to implement a **Product Listing Page (PLP)** as per the Figma design provided by **Appscrip**.
This project demonstrates proficiency in:

- Converting Figma UI into functional, responsive React code
- Understanding and implementing **Server-Side Rendering (SSR)** concepts
- Applying **SEO optimizations** (meta tags, schema, alt text, etc.)
- Maintaining clean and scalable **React architecture**

⚙️ 2. Technology Stack

Category	Technology Used
Frontend Framework	React.js (via Create React App)
Language	HTML5, CSS3, JavaScript (ES6+)
API	FakeStoreAPI
Version Control	Git + GitHub
Hosting / Deployment	Netlify
Build Tool	React Scripts (Webpack under the hood)
Code Editor	Visual Studio Code
CSS Methodology	Component-level modular CSS
Browser Compatibility	Chrome, Edge, Safari, Firefox

🏗️ 3. Project Structure

```
Appscrip-task-Bonaboina_Chithra_Gowtham/

├── public/
│   ├── index.html              # HTML template with SEO meta and schema
│   ├── favicon.ico
│   └── assets/
│       └── icons/              # SVG and image assets used throughout
│
├── src/
│   ├── assets/
│   │   └── icons/              # SVGs imported in components
│   │
│   ├── components/             # Reusable, modular React components
│   │   ├── Header.js / Header.css
│   │   ├── Filters.js / Filters.css
│   │   ├── ProductCard.js / ProductCard.css
│   │   ├── ProductGrid.js / ProductGrid.css
│   │   └── Footer.js / Footer.css
│   │
│   ├── App.js                  # Root component assembling UI
│   ├── App.css                 # Global styles
│   └── index.js                # Entry point
│
├── package.json                # Scripts, dependencies, build settings
├── .gitignore
├── .nvmrc                      # Forces Node v18 on Netlify
└── README.md                   # Project documentation
```

🧩 4. Component Breakdown

🔹 Header Component (`Header.js`)

Features:

- Company logo (SVG)
- Navigation icons for search, wishlist, bag, and profile
- Responsive layout: collapses into minimal icons for mobile view
- Fixed position with shadow for better UX

Figma References:

- Matched spacing, alignment, and icon proportions
- Implemented hover effects and icon tooltips

🔹 Filters Component (`Filters.js`)

Features:

- Sidebar filters for categories, price, and color
- “Hide Filter” toggle to collapse sidebar (for mobile responsiveness)
- “Recommended ▼” dropdown aligned to the **right** as per Figma
- Fully responsive using Flexbox and Grid
- Filter icons integrated with `src/assets/icons/`

Logic:

- Filters are currently static (UI-based)
- Designed to be API-ready for future dynamic filtering

🔹 ProductGrid Component (`ProductGrid.js`)

Features:

- Fetches product data from **FakeStoreAPI**
- Displays products in a responsive grid (3, 2, or 1 column)
- Each product rendered via the `ProductCard` component
- Loading and error state management included

API Endpoint:

`https://fakestoreapi.com/products`

🔹 ProductCard Component (`ProductCard.js`)

Features:

- Displays product image, name, and price
- Heart (wishlist) icon overlay on top right
- SEO-compliant image `alt` text
- Clean hover transition

Responsive Design:

- Scales images proportionally
- Adjusts text font-size for mobile screens

🔹 Footer Component (`Footer.js`)

Features:

- Footer sections: About, Help, Shop, and Contact
- Newsletter subscription input
- Social media and payment icons (PayPal, ApplePay, GPay, etc.)
- Accordion-style sections on mobile view
- SEO-friendly links and aria-labels

🌐 5. Responsiveness (Mobile & Tablet)

Device	Layout
Desktop (≥1024px)	3-column product grid + visible sidebar filters
Tablet (768–1023px)	2-column grid, collapsible filters
Mobile (≤767px)	1-column grid, filters hidden by default, accordion footer

Used CSS Grid and Flexbox with media queries:

```
@media (max-width: 768px) {
  .product-grid {
    grid-template-columns: repeat(2, 1fr);
  }
}
@media (max-width: 480px) {
  .product-grid {
    grid-template-columns: 1fr;
  }
}
```

🔍 6. SEO Optimization

- **Title:** `Metta Muse | Discover Our Products`

Meta Description:

```
<meta name="description" content="Browse our exclusive collection of products on Metta Muse. Filter by category, price, and more.">
```

- **Heading Structure:**
    - H1: Product Listing
    - H2: Filters, Recommendations

Schema.org Markup:

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ItemList",
  "itemListElement": []
}
</script>
```

- **Image Alt Text:** All product images use `alt="Product: {title}"`

🚀 7. Deployment Instructions (Netlify)

Step 1: Push to GitHub

Make sure your final changes are committed:

```
git add .

git commit -m "Final build for Netlify"

git push origin main
```

Step 2: Connect to Netlify

1. Go to https://app.netlify.com/
2. Click **“Add new site → Import from GitHub”**
3. Select your repository (`Appscrip-task-Bonaboina_Chithra_Gowtham`)
4. Configure:
    - **Build Command:** `npm run build`
    - **Publish Directory:** `build`
    - **Node Version:** `18` (via `.nvmrc` or Netlify environment)
5. Click **Deploy Site**

Your app will build and deploy automatically on every GitHub push.

🧠 8. Learning & Best Practices Applied

- Followed **component-based architecture** in React
- Maintained **minimal DOM nesting**
- Used **semantic HTML** elements (`<header>`, `<main>`, `<section>`, `<footer>`)
- Avoided heavy third-party libraries (no Bootstrap, only React)
- Wrote modular, reusable CSS
- Used **Git version control** with clean commit history

🗒️ 9. Evaluation Checklist (Matches Appscrip Requirements)

Evaluation Criterion	Status
HTML & CSS Implementation	✅ Completed
React Functional Page	✅ Completed
SSR Awareness	✅ Documented
Responsive for Mobile/Tablet	✅ Completed
Code Structure & Naming Convention	✅ Clean & Modular
Minimum Pre-built Packages	✅ Used only React & Axios
SEO (Title, Meta, Schema, Alt text)	✅ Implemented
Public Hosting (Netlify)	✅ Live
GitHub Repository (Public)	✅ Available
Mock API Integration	✅ Using FakeStoreAPI

🧰 10. Future Improvements

- Implement real-time API filtering
- Add sorting and pagination logic
- Integrate cart functionality and checkout flow
- Improve Lighthouse SEO and accessibility scores
- Migrate to Next.js for full SSR capability

👨‍💻 Author

Name: Bonaboina Chithra Gowtham
Email: bonaboinagowtham@gmail.com
GitHub: https://github.com/Tinku070
Deployed App: https://appscrit-task-gowtham.netlify.app/