



Mother's Day Tribute Blog Challenge (React.js / Next.js)

Objective:

Develop a Mother's Day Tribute Blog based on the provided [Figma Design](#).

The project simulates building a mini-blog website — testing frontend skills in UI replication, routing, dynamic pages, state handling, and responsiveness.

Requirements:

Pages and Features:

1. Home Page (/)

- Implement the **Hero Section**:
 - Carousel displaying featured tribute stories (3–5 stories).
 - Include navigation arrows (next, previous).
- **Explore by Category Section**:
 - Cards for different categories (e.g., Stories, Health, Inspiration).
 - Clicking a category should filter the Recent Articles section (bonus).
- **Recent Articles Section**:
 - Cards showing:
 - Image
 - Title
 - Short excerpt (2–3 lines)
 - Reading time
 - Category tag
 - 'Read More' button linking to individual story pages.



- **Sidebar** (Static Content):
 - Author Profile (static).
 - Destinations or Highlights (static).

2. Story Detail Page (/articles/:id)

- Dynamic routing: each article opens a detailed view page.
- Display:
 - Full article content
 - Title, Author Name, Date, Category
 - Reading time

3. Search Functionality (Bonus)

- A search bar that filters articles by Title or Excerpt (client-side search).

4. Category Filter (Bonus)

- Clicking a category filters the articles dynamically.

Data Handling:

- Use a **local JSON file** to store articles.
- Each article object should include:
 - ID
 - Title
 - Excerpt
 - Full Content
 - Author Name
 - Submission Date
 - Category



-
- Reading Time (in minutes)
 - Image URL (use placeholders if needed)

No backend or API setup is needed.

Technical Guidelines:

- Use **Next.js** (preferred) or **React.js with React Router**.
- Use **modular, clean CSS** (Tailwind CSS preferred but not mandatory).
- Ensure **full responsiveness** (Desktop, Tablet, Mobile).
- Implement a **working carousel** (basic next/prev functionality).
- **State management** can be basic (props, Context API if necessary).

Deliverables:

- GitHub Repository:
 - Clear folder structure.
 - README with:
 - Project Overview
 - How to run locally
 - Features implemented
- Live deployment (Vercel, Netlify) — optional but recommended.

Evaluation Criteria:

- UI Fidelity (how closely it matches the Figma design)
- Code structure and maintainability
- Functionality completeness
- Responsiveness across devices
- Creativity (extra features like search, theme toggle, pagination, etc.)



Timeline:

- **Duration:** 2 days from challenge start date.

Notes:

- Students must use the provided Figma file as reference.
- No backend is required.
- Focus on clean, production-quality frontend code.

Submission Process:

Please follow the steps below to complete your submission:

1. Push Your Code to GitHub

- Create a **public GitHub repository**.
- Your repository should contain:
 - A clear folder structure.
 - A README.md file that includes:
 - Brief project description
 - Setup instructions
 - Features implemented
 - Live deployed link (if available)

2. Deploy Your Project (Recommended)

- Deploy your project using platforms like Vercel or Netlify.
- Share the deployment link in your form submission.

3. Submit the Google Form

- After pushing your code and deploying your project, submit your final entry here:

👉 [Submit Your Project Here](#)