Andrew Lenhart | Software Developer

Inquiring Role: React Developer
Location: Greensboro/Kernersville, North Carolina, USA
Email: andrew.code21@gmail.com
Github: https://github.com/lenharta
CodeSandbox: https://codesandbox.io/u/lenharta
Phone: (336) 269-4240

Summary

A motivated, self-taught, developer based in Greensboro, North Carolina with 3+ years of experience building beautiful, accessible, and human-centered digital experience utilizing tools such as TypeScript, JavaScript, ReactJS, Redux, SASS/SCSS, NodeJS, ExpressJS, MySQL, GraphQL, NextJS, GatsbyJS, ViteJS, WebpackJS, Babel.

Technologies

Programming Languages: TypeScript, JavaScript, HTML, CSS/SCSS
Version Control/Repos: Git, Github, NVM.
SDLC Method/Software: Agile, Scrum, Waterfall, JIRA.
Testing: Jest, React Testing Library.
Database: MySQL, GraphQL, Prima, Firebase, AWS.
Other: JWT, Express, NodeJS, Redux, Helmet, Framer Motion, React Router, Axios Client.

I've provided a list of current technologies that I am using day-to-day for my projects. To give you a broader overview, I like to keep a running list of the technologies I have used over the years. **See more at the bottom of the document.**

Experience

Hatch.co

Duration: Mar. 2021 - Dec. 2022
Position: React Developer | Hatch Online Store

Hatch is a technology company, Headquartered in Winston-Salem, North Carolina that develops devices to help users sleep better at night and create healthy sleep habits. Devices such as Hatch Restore is an advanced sleep companion that is part sound machine, part night light, and Alarm clock that is controlled through WIFI through its companion app. At Hatch, I help build

and maintain the web-based online store that allows customers to browse, compare, and purchase both physical and digital products.

Responsibilities

- Creating and maintaining a dynamic SPA utilizing JavaScript, TypeScript, React, Redux, React Router, CSS3, SCSS and GraphQL, Webpack, and Babel.
- Working with product managers and designers to flesh out project requirements before implementation.
- Collaborate with developers and designers to produce common components to enforce brand consistency across platforms.
- Working in an Agile environment to promptly deliver features to our internal stakeholders.
- Perform version control for efficiency with colleague developers using Git.
- Built modules for high performance data visualization dashboard using advanced React concepts such as higher order component (HOC)
- RESTful web services in collaboration with Contentful CMS and GraphQL.
- Developing custom components in collaboration with for collecting customer information and preferences for the email based Newsletter.

Hawthorne Residential Partners

Duration: Dec. 2019 - Feb. 2021
Position: React Developer | Online Resident Platform

Hawthorne Residential Partners is a National Multifamily Housing and Development company based in Greensboro, North Carolina. With Hawthorne I assisted in the development and deployment of an Online Platform for residents across many different properties creating a seamless and convenient digital experience to communicate with the leasing office. Users could schedule payments, maintenance, and appointments as well as receive/give feedback between the leasing office and its residents. I also worked on digital experiences that showcased Hawthorne properties to new residents. Working with a team of UI Designers and creating a digital experience that would allow users to explore available properties and schedule an appointment with a leasing consultant.

Responsibilities

- Developed and tested web applications utilizing JavaScript, TypeScript, jQuery, HTML5, CSS3, and Material UI.
- Developed Client-side Menus, search bars, and carousels in collaboration with internal designers.
- Utilized CSS Preprocessor SCSS to make CSS development more efficient and scalable.
- Created cross-browser compatible and responsive page layouts.

- Enriched internal UI library with components and website elements for use in cross-client projects.
- Utilized Axios Http Client to handle JSON send/receive Payloads from a client server.
- Managed version control throughout the development of the project by using Git.
- Developed a deeper understanding of WCAG and Accessibility Compliance.
- Developed a deeper knowledge of Chrome and Safari Developer tools for troubleshooting and debugging.

Pace Communications

Duration: May 2019 - Sept 2019
Position: React Developer | Various Client Projects

Pace Communications is an International Branding and Marketing Agency Based in Greensboro, North Carolina helping Clients empower their brand and grow their influence, and moving their audience to action. Helping the likes of clients such as Wells Fargo, PepsiCo, Four Seasons, Verizon, and Sysco. At Pace, I assisted in developing website elements for reuse and scalability in internal UI Libraries. I also worked closely with our designers on client projects developing custom JavaScript components for use in their web applications.

Responsibilities

- Light/Dark Theme
- Full Site Design by me
- 100% Custom UI, No Libraries
- Built utilizing Javascript, Typescript, React, and Gatsby
- Use of Styled-Components for Styling and Support reusability
- Custom Animations with Framer Motion
- Use of Built-in and Custom react hooks
- Search Technologies input
- Utilization of React Testing Library for unit tests
- Use of HOC for component reuse and scalability
- Parallax Effect Heroes, Banners
- Static Page and Image generation by Gatsby with GraphQL
- Full site SEO Optimization utilizing React Helmet
- Use of WCAG Standards for accessibility
- Completely Responsive across Devices
- Semantic HTML5 Elements
- CSS Support Checking and fallback for cross-browser compatibility
- Heuristic and Gestalt UI/UI Design Principles
- Accesses Local Storage for saving theme preference
- Deployed project via GitHub Pages

Personal Projects

CORTX To-do App

Live Demo: https://cortx-app-v1.netlify.app

CortX is a simple to-do app built with React. That allows users to create, read, filter, sort,edit, and delete Todos. Completed to showcase abilities in creating organized Dynamic web pages that interact with external databases and demonstrate use of basic built-in React hooks, React file structure, components, JSX, Vanilla CSS3, BEM Class naming structure, responsive development and design, and efficiency in building beautiful web pages.

Features

- Full Site Design by me
- 100% Custom UI, No Libraries
- Built with Javascript, React, CSS3, SCSS, HTML5
- React Context - API Data
- Powered by Google Firebase Realtime Database
- Dynamic Routing Utilizing React Router V6
- Semantic HTML5 Elements
- Heuristic and Gestalt UI/UI Design Principles
- Completely Responsive across Devices
- Axios HTTP Client
- Use of Higher Order Components for scalability
- Custom Design utilizing CSS3 with SCSS Modules
- Use of WCAG Standards for accessibility, ARIA, reduced motion, keyboard navigation
- Deployed Project via Netlify

Portfolio v1

Live Demo: https://lenharta.netlify.app

The first iteration of my personal portfolio to showcase completed projects. Built using Javascript, React and Styled-Components; it features many animations, a custom loader, the technology stack I worked with at the time, links to my social accounts, a biography about myself, and statistics of my GitHub profile. This was a fun project because it was a great chance to not only express myself while showing off my skills but to also build a time capsule of code to revisit in the future to show myself progression in my abilities as a software developer.

Features

- Built using JavaScript, React, HTML5, CSS3
- Built 100% Custom no, use of UI Libraries
- 100% Custom design created by me

- Styled-Components for scalability of Styling
- Custom Hook for reveal/disappearing Desktop Navigation on user scroll Direction
- Animated Mobile Menu with useOnClickOutside Hook for closing menu based on ref container
- Stateful Accordion component in bio
- Icon HOC for reusability of icon components and custom SVG illustrations
- Animated Loader utilizing AnimeJS
- Scroll animations with RevealScroll and CSSTransition Group
- Prop typing with React PropTypes
- Deployed Project via Netlify

Disney+ Web App Clone

Live Demo: https://disneyplus-clone-82c31.web.app

This project was inspired by the Disney+ Streaming Web Application. I wanted to create a clone of their application to gather experience with some technologies I was exploring at the time. This App is built with a combination of React, React Router, Redux State Management, and Google Firebase. It offers authentication and sign in with a Google Account and full a directory of current available Disney shows and movies with previews. A complete copy of the Disney Web App thanks to UI Designers at Disney and the use of Styled-Components. This was truly a fun project to work on!

Features

- Built with Javascript, React, CSS3, HTML5
- Data on Movies and Shows stored with Firebase FireStore
- React Redux Toolkit Global State Management for User and Movie features.
- Authentication provided by Firebase and Google Authentication
- Dynamic Routing with React Router
- Scalable styling structure with utilizing Styled-Components
- Featured Movies and Shows carousel provided by react-slick library.
- Use of Higher Order Components. Built-in React Hooks, asynchronous requests.
- Deployed Project via Firebase

OpenAi GPT-3 API Promotion

Live Demo: https://gpt3-mockup.netlify.app

Showcases' abilities in creating organized static web pages and demonstrate use of basic built-in ReactJS hooks, React file structure, components, JSX, Vanilla CSS3, BEM Class naming structure, responsive development and design, and efficiency in building beautiful web pages.

Features

- Built using JavaScript, React, CSS3, HTML5, JSX
- BEM (Block - Element - Modifier) CSS naming
- Use of Semantic HTML5 elements
- Passing props Demonstration
- Clean file structuring
- Deployed Project via Netlify

Personal Portfolio v2

Live Demo: Work in Progress

The second iteration of my personal portfolio website to showcase completed projects and share information about myself and what I can offer clients and employers. This website is built with JavaScript, TypeScript, and ReactJS. It includes a custom loading screen, Light/Dark Theme, parallax effects, asymmetrical design, and some amazing animations. In version two of my portfolio I tried to show my growth into a more diverse, creative, and robust developer. I completed this project recently and It has been very interesting to compare it to its predecessor (referenced later in this section) and see the personal growth I have acquired as a developer. It has been a ton of fun developing and expressing myself with these projects and I hope to do more versions in the future to keep a time capsule of code over my career!

Features

- Light/Dark Theme
- Full Site Design by me
- 100% Custom UI, No Libraries
- Built utilizing Javascript, Typescript, React, and Gatsby
- Use of Styled-Components for Styling and Support reusability
- Custom Animations with Framer Motion
- Use of Built-in and Custom react hooks
- Search Technologies input
- Utilization of React Testing Library for unit tests
- Use of HOC for component reuse and scalability
- Parallax Effect Heroes, Banners
- Static Page and Image generation by Gatsby with GraphQL
- Full site SEO Optimization utilizing React Helmet
- Use of WCAG Standards for accessibility
- Completely Responsive across Devices
- Semantic HTML5 Elements
- CSS Support Checking and fallback for cross-browser compatibility
- Heuristic and Gestalt UI/UI Design Principles
- Accesses Local Storage for saving theme preference
- Deployed project via GitHub Pages

Contact Information

Method	Value
Phone	(336) 269-4240
Location	Greensboro, North Carolina USA
Email	andrew.code21@gmail.com
Github (url)	https://github.com/lenharta
Portfolio (url)	https://lenharta.netlify.app

Technologies Table

Technology Name	URL
ReactJS	https://react.dev
TypeScript	https://www.typescriptlang.org
Javascript ES7+	https://developer.mozilla.org
CSS3	https://developer.mozilla.org
SCSS	https://sass-lang.com
HTML5	https://developer.mozilla.org
ExpressJS	https://expressjs.com
GatsbyJS	https://www.gatsbyjs.com
NextJS	https://nextjs.org
NodeJS	https://nodejs.org/en
Styled Components	https://styled-components.com
MySQL	https://www.mysql.com
GraphQL	https://graphql.org
ExpressJS	https://expressjs.com
Webpack	https://webpack.js.org
Babel	https://babeljs.io
Git	https://git-scm.com

Github	https://github.com
Jest (Testing)	https://jestjs.io
Jira (Agile Software)	https://www.atlassian.com/software/jira
Jenkins (Automation)	https://www.jenkins.io
React Test Library (Testing)	https://testing-library.com/docs
AWS (Amazon Web Services)	https://docs.aws.amazon.com
Google Firebase (Web Services)	https://firebase.google.com/docs
Docker (Distributed Containers)	https://docs.docker.com
Kubernetes (Container Orchestration)	https://kubernetes.io
Shopify Storefronts (Shopify Store)	https://shopify.dev/docs
Framer Motion (Animations)	https://www.framer.com/motion
Axios (HTTP Client Manager)	https://axios-http.com/docs/intro
Redux (State Management)	https://redux.js.org
Xstate (State Management)	https://xstate.js.org
Jotai (State Management)	https://jotai.org
MaterialUI (React Component Library)	https://material-ui.com/
Babel (JS Compiler)	https://babeljs.io/docs/en/
Webpack	https://webpack.js.org/
Figma	https://www.figma.com/
Netlify	https://docs.netlify.com/
Firebase	https://firebase.google.com/docs/
React Testing Library	https://testing-library.com/docs/react-testing-library/intro
JestJS	https://jestjs.io/docs/en/getting-started.html