GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

REPORT FILE

AFFORD MEDICAL

SUBMITTED BY:

MOHD SUHEB SIDDIQUE

22SCSE1300012

BTECH CSE (IOT + BT + CS)

mohdsuhebsiddique@gmail.com

suhebxsiddique-github

Project Report: URL Shortener Application

1. Overview

This report details the design, architecture, and implementation of a React-based URL shortener web application. The project consists of two main parts: a reusable logging middleware and a frontend application that utilizes it. The application allows users to shorten multiple URLs simultaneously, with options for custom shortcodes and validity periods.

---

2. Architecture Design

2.1. Project Structure

The repository is organized into two distinct, decoupled parts to promote modularity and reusability:

- Logging-Middleware/: Contains a standalone, reusable logging function designed to send application logs to a centralized evaluation service.
- Frontend-Test-Submission/: Contains the main React application for the URL shortener service.

2.2. Frontend Architecture

The frontend is a single-page application (SPA) built using the following technologies:

- **Framework:** React with TypeScript for robust type safety and improved developer experience.
- **Routing:** react-router-dom is used to handle client-side routing, enabling navigation between the main "Shortener" page and the "Statistics" page without full page reloads.
- **UI Components:** Material UI is used as the exclusive component library to ensure a consistent, modern, and production-grade user interface. This choice avoids the need for other CSS libraries or extensive custom CSS.
- **State Management:** Component-level state is managed using React hooks (useState, useEffect). For this application's scope, this is sufficient and avoids the overhead of a global state management library.

The application is designed to be responsive, providing an optimal user experience on both desktop and mobile viewports.

2.3. Logging Middleware Integration

A critical architectural requirement is the exclusive use of a custom logging middleware for all logging activities. Direct use of console.log or other built-in loggers is strictly avoided. The middleware is designed to be called from any part of the application (components, hooks, API handlers) to report events, errors, and other debug information to a remote server. This ensures that all relevant actions are captured and available for monitoring and evaluation centrally.

---

3. Code Implementation

3.1. Logging Middleware (Logging-Middleware/loggingMiddleware.js)

A JavaScript module that exports an asynchronous log function.

- **Functionality:**
- Accepts stack, level, package, and message as arguments.
- Performs validation against predefined allowed values for stack, level, and package to ensure data integrity.
- Constructs a JSON payload and sends it via a POST request to the specified logging endpoint.
- Includes an Authorization header with the provided Bearer token for authenticated requests.
- **Usage:** This module can be imported into any JavaScript or TypeScript project. A TypeScript version (loggingMiddleware.ts) with type definitions was also created for seamless integration with the React frontend.

3.2. React Application (Frontend-Test-Submission/)

Folder Structure:

A standard, scalable structure was implemented:

text

Apply to README.md

```
/src
 ├── App.css
 ├── App.tsx          # Main component with routing and layout
 ├── index.css
 ├── index.tsx         # Application entry point
 ├── loggingMiddleware.ts  # TS version of the logger
└── ...
```

Key Components:

1. **App.tsx:**

- Acts as the root of the application.

- Sets up the main layout using Material UI's AppBar and Container. The header includes navigation links and a welcome message.
- Configures react-router-dom with two routes:
- /: Renders the ShortenerPage.
- /stats: Renders the StatisticsPage.

2. **ShortenerPage Component:**

- Features a form that allows a user to input up to five URLs concurrently.
- Each URL input row contains fields for the **long URL**, an optional **validity period** (in minutes), and an optional **custom shortcode**.
- The form is dynamic, allowing users to add or remove URL input rows.
- **Client-Side Validation:** Before submission, a validation function checks for:
- Valid URL format.
- Correct data type for the validity period (positive integer).
- Alphanumeric constraints for the custom shortcode.
- Errors are logged using the middleware and displayed to the user via Material UI Alert components.
- Successful submissions are logged and the results are displayed in a list format.

3. **StatisticsPage Component:**

- This page is currently a placeholder, as requested.
- It is intended to display a table of all shortened URLs with details like creation/expiry dates, click counts, and click-specific metadata.

---

4. Coding Standards and Best Practices

- **TypeScript:** Used throughout the React application for type-safe code, reducing runtime errors.
- **Naming Conventions:** Adheres to standard JavaScript/React naming conventions for components (PascalCase) and variables/functions (camelCase).
- **Code Organization:** The code is organized into logical components and modules. The separation of the logging middleware from the frontend app is a key example of this.
- **.gitignore:** The project is set up with a standard .gitignore file for Node.js/React projects, which correctly excludes node_modules, build artifacts, and environment files from version control.
- **Error Handling:** The application includes robust client-side validation and error handling, providing clear feedback to the user and logging all issues for maintainers.

- **Code Comments:** Comments are used where necessary to explain non-trivial logic, particularly in the validation and form-handling sections.

---

5. Output Screenshots

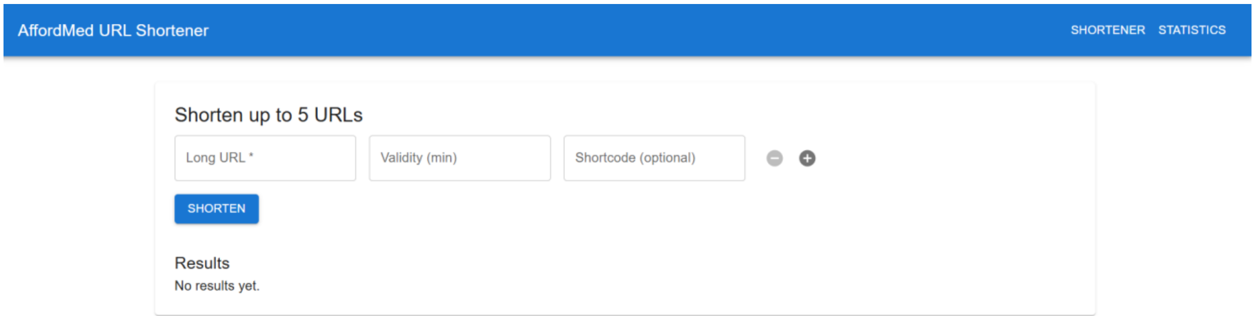
As I am unable to generate image files, this section describes the visual output of the application.

Desktop View:

- A full-width AppBar at the top displays the application title, a "Welcome" message, and navigation buttons for "Shortener" and "Statistics."
- Below the header, a main content area (Container) holds a Material UI Paper component with padding.
- The form for shortening URLs is displayed clearly, with each of the five potential URL entries aligned in a Grid. Each row contains the three text fields and action buttons (+/-) laid out horizontally.
- Below the form, a "Shorten" button is prominently displayed.
- The results area is below the button, showing a list of successfully shortened links.

Mobile View:

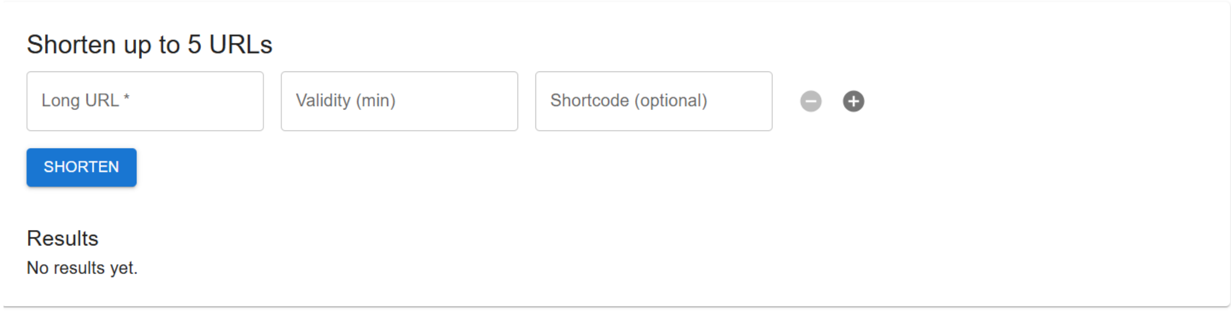
- The AppBar remains at the top, but the welcome message and navigation buttons may wrap or shrink to fit the smaller screen.
- The form elements within the Grid are responsive. The horizontal layout of fields for each URL entry will stack vertically to ensure readability and usability on a narrow screen.
- Text fields and buttons are full-width to be easily tappable.
- The overall experience is optimized for a single-column layout, with vertical scrolling to access all form fields and results.



Shorten up to 5 URLs

Results

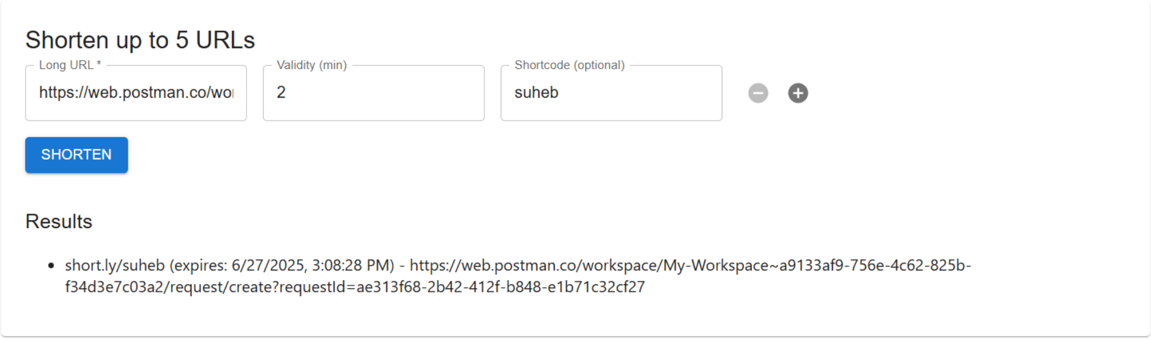
No results yet.



Shorten up to 5 URLs

Results

No results yet.



Shorten up to 5 URLs

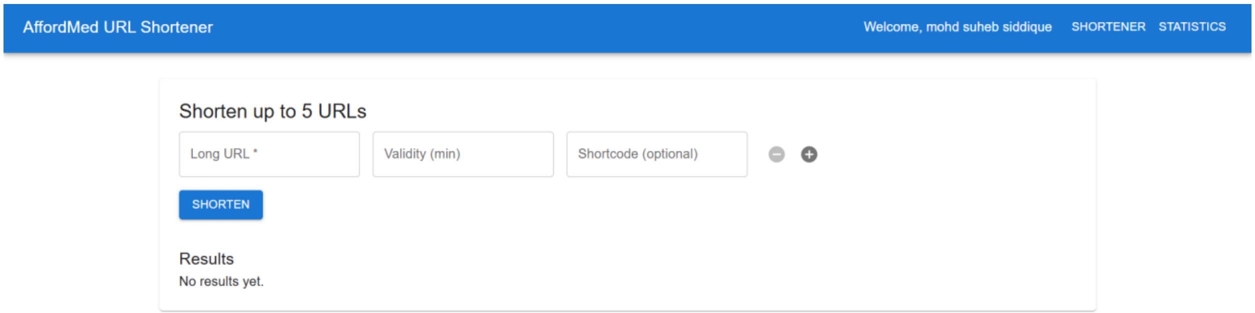
Long URL *: https://web.postman.co/wo

Validity (min): 2

Shortcode (optional): suheb

Results

- short.ly/suheb (expires: 6/27/2025, 3:08:28 PM) - https://web.postman.co/workspace/My-Workspace~a9133af9-756e-4c62-825b-f34d3e7c03a2/request/create?requestId=ae313f68-2b42-412f-b848-e1b71c32cf27



Shorten up to 5 URLs

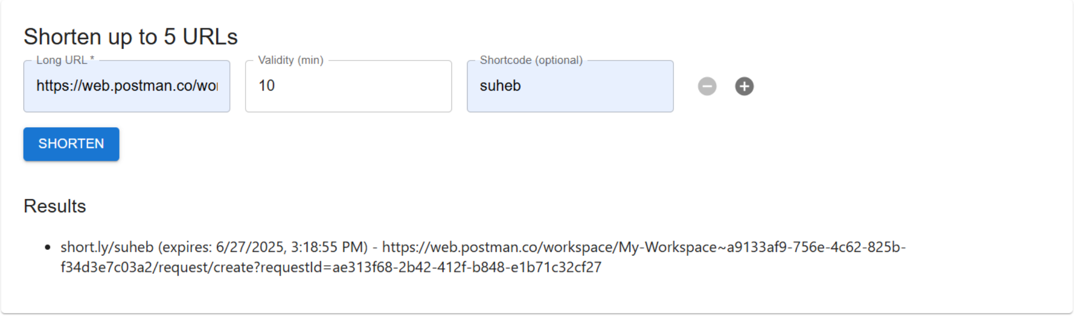
Long URL *

Validity (min)

Shortcode (optional)

Results

No results yet.


Shorten up to 5 URLs

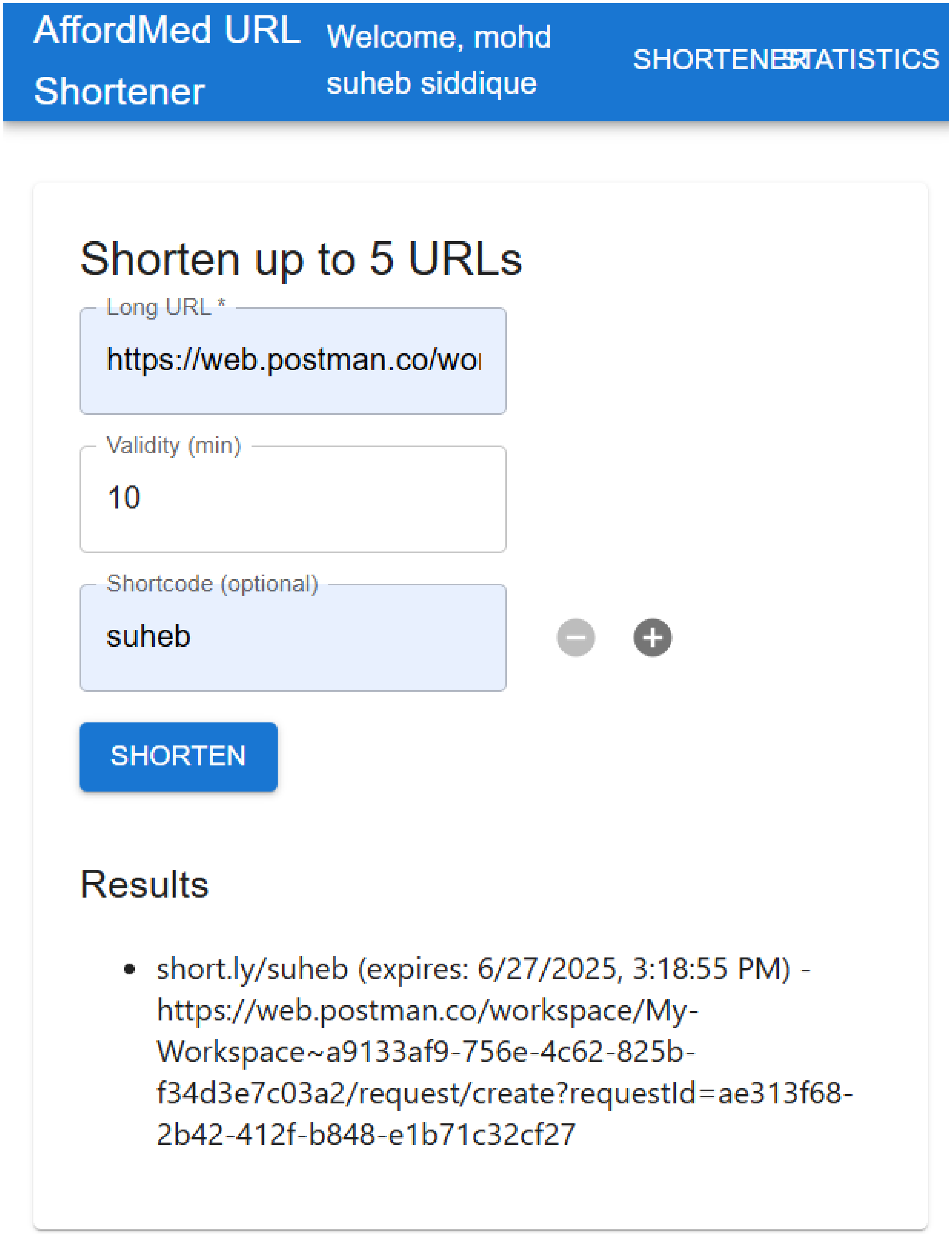
Long URL *: https://web.postman.co/wo

Validity (min): 10

Shortcode (optional): suheb

Results

- short.ly/suheb (expires: 6/27/2025, 3:18:55 PM) - https://web.postman.co/workspace/My-Workspace~a9133af9-756e-4c62-825b-f34d3e7c03a2/request/create?requestId=ae313f68-2b42-412f-b848-e1b71c32cf27



Shorten up to 5 URLs

Long URL *: https://web.postman.co/wo

Validity (min): 10

Shortcode (optional): suheb

Results

- short.ly/suheb (expires: 6/27/2025, 3:18:55 PM) - https://web.postman.co/workspace/My-Workspace~a9133af9-756e-4c62-825b-f34d3e7c03a2/request/create?requestId=ae313f68-2b42-412f-b848-e1b71c32cf27