Backend Engineer Assessment

Introduction

At Topup Mama we take great pride in tackling complex problems and creating solutions that are innovative and impactful. To deliver on that promise, we follow best practices in developing applications, e.g, having a well-defined directory structure, writing clean modular and microservices code, and driving development via tests.

Given our bar for quality, we have high expectations from anyone wishing to join us. Please use all the allocated time to craft a solution that you can be proud of and help us evaluate you in the best light.

Objective

This task is designed to provide an opportunity for you to demonstrate general restful API development knowledge in the sense that you:

- Document your solution architecture/workflow
- Write clean, structured, readable, and maintainable code.
- Create simple application components and building blocks.
- Understand fetching, transforming, and aggregating data from external APIs.
- Maintain a well-designed application state.
- Craft a pleasant API consumer experience.
- Consume your designed API endpoint using a frontend technology (Angular or React preferably). Although, you will not be graded on the UI but only the backend code/structure.
- Document your APIs with the right HTTP verbs

Task

Create a small set of rest API endpoints using Nodejs (Typescript) that can be used for listing the names of books along with their authors and comment count, adding and listing anonymous comments for a book, and getting the character list for a book.

General requirements

- The application should have basic documentation that lists available endpoints and methods along with their request and response signatures.
- The exact API design in terms of the total number of endpoints and HTTP verbs is up to you.
- Keep your application source code on a public repository.
- Provide a live demo URL, you could spin up a virtual server on AWS, Digital Ocean, or a Heroku instance.

Data requirements

- The book data should be fetched online from https://anapioficeandfire.com/
- Book names in the book list endpoint should be sorted by release date from earliest to newest and each book should be listed along with authors and count of comments.
- Comments should be stored in a SQL database.
- Error responses should be returned in case of errors

Character list requirements

- The endpoint should accept sort parameters to sort by one of name, gender, or age in ascending or descending order.
- The endpoint should also accept a filter parameter to filter by gender.
- The response should also return metadata that contains the total number of characters that match the criteria along with the total age of the characters that match the criteria.
- The total height should be provided both in months and in years. For instance, 28 months is 2.5 years.

Comment Requirements

- The comment list should be retrieved in reverse chronological order.
- Comments should be retrieved along with the public IP address of the commenter and the UTC date & time they were stored.
- Comment length should be limited to 500 characters

Task Submission

- Create a GitHub repo for each of your services
- Deploy your services to Heroku
- Document your service boot up and Heroku link in your GitHub repository