Learning Programmer

Introduction:

The idea of website named as learning programmer is to start blogs of my own related to technology and students puruing programming or those who wants to get started with it, with some added features like newsletters and storing the topics suggested by the users into a database using NodeJS, Express and MongoDB.
By medium of these blogs, I am targeting the students or more specifically students learning programming, and even the people trying to get started with programming. I'm willing I can help with daily struggles and problems; and even give them little advices in order to improve their skills and future lives.

Expected List of Features:

- List of all the blogs
  This list would include all the blogs from where the user can navigate to different blog posts and read them. This could be implemented using general HTML, CSS a nd JS.
- Section in the footer to subscribe to the email newletter
  This is a feature that I would want to be using NodeJS combined with MongoDB for creating an object everytime a user subscribes with the email, convert it to JSON and store it in MongoDB.
- Letting the users send some topics and questions for future blogs
  This is another function using NodeJS and MongoDB to store the strings sent by the user into a database.
- User rating as per each blog post
  Rating on each blog can help me understand which type of blogs users are more into. This could also be attained using MongoDB and NodeJS with some basic formulas.
- Light Mode – Dark Mode toggle
  Light mode and dark mode toggle is yet another general feature which must be present when creating a blog website for programmers since it is easy to read in low light and helps in waking up late at night. This could be done using general JS and CSS but NodeJS might come in handy for impleenting the changes to all the pages and/or setting up environment setting files.

Design and Implementation:

The REST API Specification:
The REST API , consists of three routers, subsRouter handling all the newsletter subscriptions, topicRouter handling all the topics suggested by the users and storing them to database, and the third one is ratingRouter handling the rating from 1 to 5 and comments from the users.

Database Schemas, Design and Structure:
The three Schemas subscribers.js, topics.js and ratings.js are defined as follows,
subscribers.js:
email – type is String, required is true and it must be unique
topics.js:
topics – types is String and required is true

ratings.js:

rating – type is Number and required is true
comment – type is String

Communication:
The communication between front end and back end will be as usual. The front end will input the emails, topics etc. and the data will then be stored in mongodb in form of javascript object notations.

Conclusions:
The result that I expect from my project is that it stores all the data securely into the database and the REST API to be combined with the front-end without leaving any corner cases that might cause bugs and users experience a bug free experience.

References:

- https://breakingintostartups.com/
- https://www.miss-thrifty.co.uk/
- https://solopreneurhour.com/
- https://www.sidehustlenation.com/
- https://knowtechie.com/

(These are the websites from where I took my inspirations for features and content)