



Backend Task

Task:

Develop a Basic Backend API for the "Collaborative and fun Story Creator" Web Application

Create a simple but functional backend API to support the core features of the Collaborative Story Creator frontend application.

You are free to use any language (Nodejs, Python, etc ..)

- Ideally, we would prefer Node.js or Python as it's easier to run.
- There are no restrictions on the frameworks/libraries to use. Choose whichever seems best suited for the task.

API Details:

1. Authentication Endpoints:

- User Registration
- User login
- Get current user info

Note: Treat it as a mock authentication service and accept any username/password.

2. Story Endpoints:

The following Endpoint should be protected

- Create a new story
- Get the list of stories
- Get a specific story
- Add a contribution to a story

Note: any user can add new text to an existing story, The Story is like a fun game, the first user starts writing a story with 2 lines, the second user tries to add 2 more lines, and after 4 contributions, you get a small story.

Technical Requirements:

- Use MongoDB or Database of your choice
- Create simple schemas for Users and Stories
- Implement basic JWT-based authentication
- Use encryption for password
- Follow RESTful principles
- Implement basic error handling
- Implement basic CORS configuration

Important:

- Attach API documentation with a request that can be executed directly. You can use Postman Collection, Swagger, etc. for documentation.

Data Models:**User Model:**

Fields: id, username, email, password (hashed)

Story Model:

Fields: id, title, contributions (array of objects with userId and content), createdBy

Additional Features:

Environment Configuration: managing environment variables properly

Basic Validation: Implement simple input validation for requests

Bonus: Extra points for attempting these.

1. Export Story in Image or PDF format
When the endpoint hit. It takes story data and creates Images or PDFs out of Story data. and stores files in storage.
(should be asynchronous and should not block other requests)
2. Write all backend code in TypeScript.

Things we are interested in:

Proper implementation of basic authentication
Correct implementation of CRUD operations for stories
Basic error handling and input validation
Code organisation and readability
Functionality of API endpoints
Git code management
API Documentation

How to submit:

Please Email your submission to the following address.

To: krushit.dudhat@redsoftware.in, sanket.debnath@redsoftware.in

Cc: jigar.chauhan@redsoftware.in

Subject: <Your Name> - Backend Task

Along with:

1. Google Drive link of your source code zip (Be careful of the sharing permission)
2. Hosted URL (Optional) [Netlify, Heroku, etc.]
3. Postman collection

Please note: For any clarifications contact: krushit.dudhat@redsoftware.in or sanket.debnath@redsoftware.in

Important:

We condemn plagiarism. Please maintain the dignity and originality of your work. If we suspect any attempt towards copying, we will disqualify the submission.

Also, keep your code on GitHub. The repo must be private and share access to krushit.dudhat@redsoftware.in, sanket.debnath@redsoftware.in