

# Internship assignment

24<sup>th</sup> July 2023

## OVERVIEW

You can pick any one of the two assignments (backend and front end) given below. Feel free to push the code to your github and submit the link to [pooja@triveous.com](mailto:pooja@triveous.com) even if it is incomplete. We just want to evaluate your problem solving approach. Also, you can add your creative touch wherever you wish to

## Table of contents

[OVERVIEW](#)

[GOAL](#)

[FRONT-END ASSIGNMENT: “News App with Firebase Integration”](#)

[Requirements:](#)

[BACKEND ASSIGNMENT: “Ecommerce API with Node js”](#)

[Requirements:](#)

## GOAL

Submit atleast one assignment. You front end assignment should either run on browser (web app) or should be installable on mobile phone (flutter app). And/Or your backend assignment should be callable from any rest client eg. Postman

## FRONT-END ASSIGNMENT: “News App with Firebase Integration”

Build a news reader app using **Next js** or **Flutter**

### Requirements:

1. User Registration and Login: Users should be able to create accounts and log in to access personalized news preferences. Use Firebase for authentication
2. News API Integration: Use an open-source news API (e.g., <https://newsapi.org/> or any other source) to fetch the latest news articles and display them in a list view (for flutter app) and two columns of list view (for web app).

- 
3. Grid view: Add a toggle button to show the news items in a grid view. UI styling can be of your choice here
  4. News Detail View: When users tap on a news article item, it should show a detailed view of the article in another full screen, including the title, description, image, and a link to the full article. You can use an in-app webview to show this if you are building a flutter app
  5. Favorite Articles: Allow users to mark articles as favorites, and store their preferences in Firebase for persistence. This can be done by tapping on a heart icon on the top-right of the list items or the details view page
  6. State management: Use some state management tools
  7. Offline Support: Implement caching mechanisms to allow users to read previously fetched news articles even when they are offline.
  8. Design and UI/UX: Pay attention to app design and create a user-friendly interface.
  9. Lastly, the app should be responsive in terms of UI and smooth in terms of UX

## **BACKEND ASSIGNMENT: “Ecommerce API with Node js”**

Build an API set to support e-commerce operations, such as product and category listing, product details, cart management, and order processing. Prefer firebase or any other suitable database to manage product/category data, user cart information, and order details. The API should also handle token management in the system

### **Requirements:**

1. API Endpoints:
  - a. Category Listing: Create an API endpoint that retrieves a list of categories.
  - b. Product Listing: Create an API endpoint that retrieves a list of products with essential details such as title, price, description, and availability, based on category Id
  - c. Product Details: Implement an endpoint that fetches the detailed information of a specific product by its ID.
  - d. Cart Management: Develop API endpoints to allow users to add products to their cart, view the cart, update quantities, and remove items from the cart.
  - e. Order Placement: Create an endpoint to handle order placement, allowing users to place an order with products from their cart.
  - f. Order History: Implement an endpoint to fetch the order history for authenticated users.
  - g. Order Details: Create an endpoint that retrieves the detailed information of a specific order by its ID.

- 
- h. A set of API to register and login the users

NOTE: Ignore any payment related APIs for simplicity. We will assume that payment is not needed for orders

2. Database Integration: Integrate MongoDB or MySQL or any other DB schema to store and manage product data, user cart information, and order details. The API should interact with DB to perform CRUD operations on products, cart items, and orders.
3. Authentication Middleware and security: Implement authentication middleware to secure sensitive API endpoints, such as cart management and order placement. Only authenticated users should be allowed to access these endpoints.
4. User Authentication: Implement user authentication using JSON Web Tokens (JWT). Users should be able to register, log in, and obtain a token to authenticate API requests.
5. Error Handling: Ensure appropriate error handling is in place, and the API returns meaningful error messages and status codes when necessary.
6. Documentation: Create documentation for the API endpoints, including details about their functionality, expected input, and output. A swagger doc is preferred
7. Rate limiting (optional): Add API rate limiting to prevent abuse and maintain server stability.

Please push the code on your github broken down into commits along with a read me file showing a brief on how to set up the project, API integration details (for front end assignment only), API doc (for backend assignment only). and any important design decisions you made during development.

Share its link to [pooja@triveous.com](mailto:pooja@triveous.com)