

Lab Exercise 7

Title: NodeJs and Express

Objectives:

Develop a product management API using Express.js that allows users to view and search for products in a catalog. The application should provide different endpoints for retrieving product information, including the ability to view all products, view details for a specific product by ID, and search for products based on query parameters.

Theory

1. What is Express.js, and why is it used in Node.js applications?

Express.js is a minimal and flexible **Node.js web application framework** that provides a robust set of features to build web and mobile applications. It simplifies the development process by providing a thin layer of fundamental web application features, without obscuring the underlying Node.js functionality. Express.js is widely used because it allows developers to handle HTTP requests, set up routing, middleware, and render dynamic HTML pages easily.

Why Express.js is Used in Node.js Applications:

2. **Routing:** Express.js provides a powerful and flexible routing mechanism for defining URL patterns and handling requests based on those patterns. It simplifies the process of managing different routes in a web app.
3. **Middleware Support:** Express allows the use of middleware to handle tasks such as request processing, logging, session handling, authentication, and more. This modular structure keeps the codebase clean and maintainable.
4. **Asynchronous and Non-blocking:** Leveraging Node.js's non-blocking, event-driven architecture, Express.js is highly scalable and can handle a large number of concurrent connections efficiently.
5. **Templating Engines:** It supports various templating engines like Pug (formerly Jade), EJS, and Handlebars, allowing the dynamic rendering of web pages.
6. **Fast Development:** Its simplicity and flexibility make rapid development of web applications possible.
7. **Integration with Databases:** Express.js can easily integrate with databases like MongoDB, MySQL, etc., to handle dynamic data

2 . How are dynamic routes handled in Express.js?

Dynamic routes in Express.js enable flexible handling of URLs with variable parts, often used for IDs, categories, or other changing values. These dynamic segments are defined with a colon (:) followed by a parameter name, such as `/products/:id`, where `:id` represents a placeholder for any value. When a user accesses `/products/45`, the value `45` is captured and stored in the `req.params` object as `req.params.id`. This allows the server to dynamically process the request based on the provided value, such as fetching product details from a database.

Multiple dynamic parameters can be defined, like `/articles/:category/:id`, capturing values like category and ID simultaneously. Express also supports optional parameters and can enforce validation using regular expressions to ensure the parameter matches a specific format.

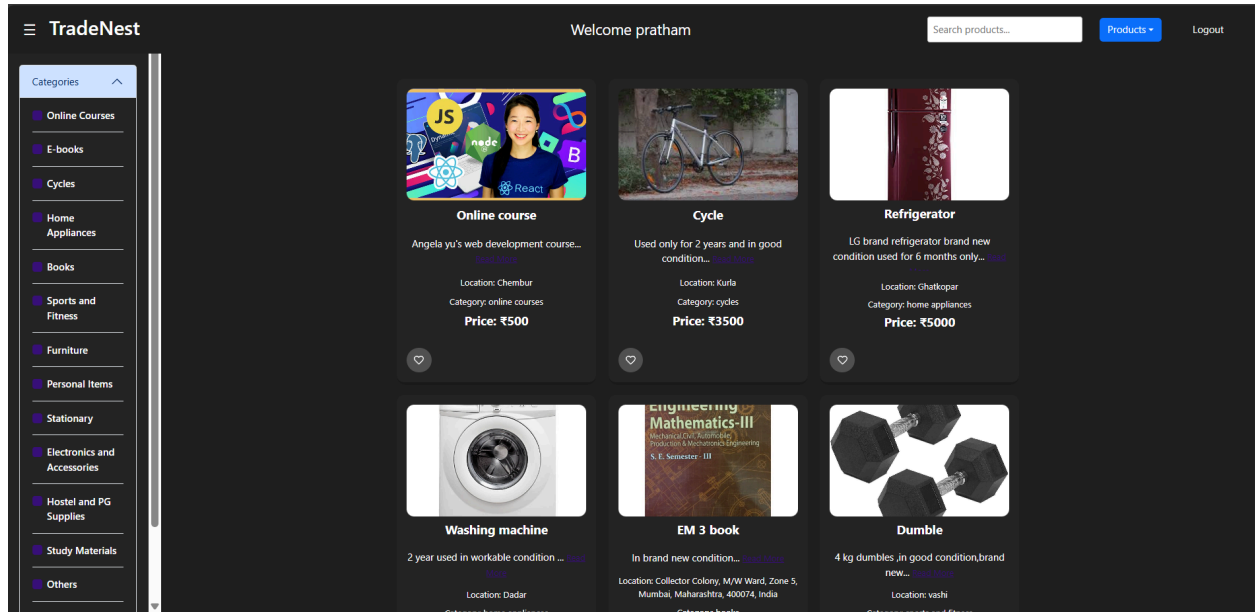
Dynamic routes make routing more efficient by reducing the need for multiple hardcoded routes, enhancing maintainability and scalability. The captured parameters are essential for operations such as retrieving, updating, or deleting specific data entries based on the provided URL values.

Hosted Page:

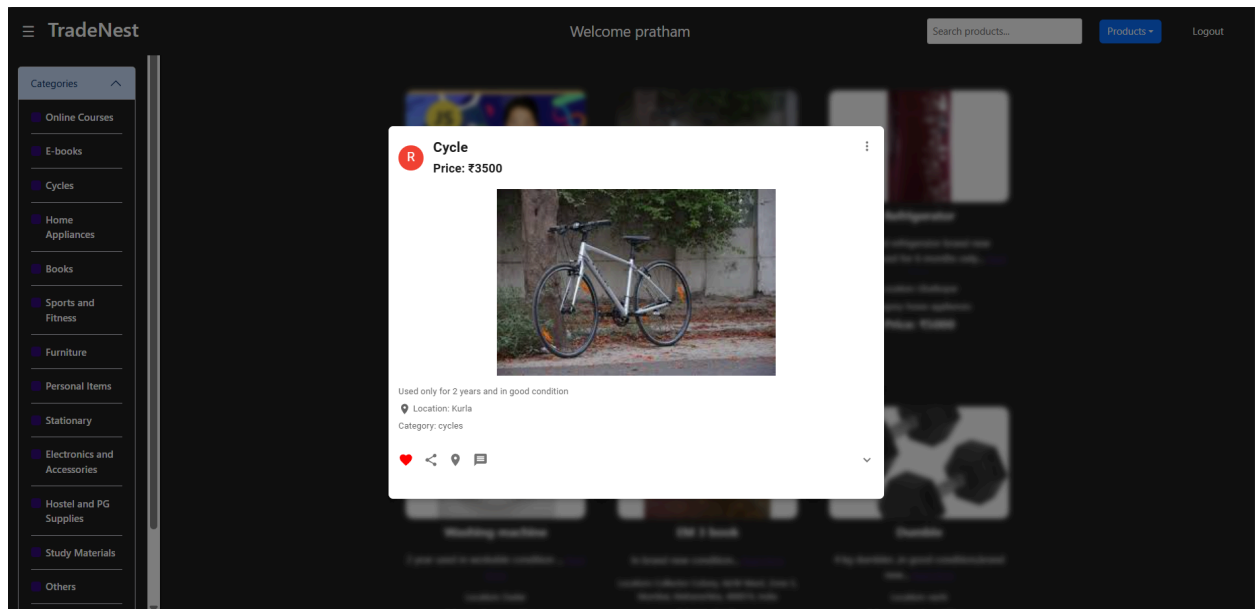
GitHub Repo for Assignment : <https://github.com/prathmzz/trade-nest>

Code with Screenshot:

rendering all products-



rendering products details-



searching products using keywords-

