NEUsed Project Functionality APIs Document

Team: Jing Shen, Yijing Liu, Yixuan Yu, Siyu Chen

Backend Tech Stack

Our team used **Graphql** to develop the schema of the data models for this project. To save the data in the **MongoDB** database, we use **Mongoose** to map and connect the data. Please see below for our backend APIs:

Query

Query in Graphql is used to retrieve data from the database. In this project, we mainly need information about the Post type. We also allow inputs for the query to return posts based on certain criterias.

Query post

Syntax: post (email, category, _id): [Post]

Input	Type	Optional	Purpose
email	String	Yes	Supporting the Your Posts page functionality, listing all posts that are created by the current signed in user
category	CategoryType	Yes	Supporting the Explore by Category pages functionalities, listing posts that are created with specific categories.
_id	Post ID	Yes	Supporting posts edit functionality

Output	Type	Optional	Purpose
posts	An array of Post type	No	Returning information field of Post type, including: - title - content - price - creator - status - image - category - email - createdAt - updatedAt

Mutation

Mutation in Graphql is used to create/update/delete data from the database. In this project, we mainly need information about the Post type.

Create post

Syntax: createPost(postInput): Post

Input	Type	Optional	User Manual Input	Input Validation
email	String(Auth data)	No	No	N/A
title	String	No	Yes	Yes

content	String	No	Yes	Yes
price	Float	No	Yes	Yes
status	Boolean	No	Yes	Yes
image	String	No	Yes	Yes
creator	String(Auth data)	No	No	N/A
category	Category Type	Yes (default value set to ‘Other’)	Yes	No
createdAt	timestamp	No	No (add in method implementation)	No

Output	Type	Optional	Purpose
Single post	Single Post type	No	Returning information field of Post type

Update post

Syntax: updatePost(postUpdateInput): Post

Input	Type	Optional	User Manual Input	Input Validation
postId	ID	No	No	Throw error if post not found
title	String	Yes	Yes	Yes
content	String	Yes	Yes	Yes

price	Float	Yes	Yes	Yes
status	Boolean	Yes	Yes	Yes
image	String	Yes	Yes	Yes
category	Category Type	Yes	Yes	No
updatedAt	timestamp	No	No (add in method implementation)	No

Output	**Type**	**Optional**	**Purpose**
Single updated post	Single Post type	No	Returning information field of Post type

Delete post

Syntax: deletePost(postId): Post

Input	**Type**	**Optional**	**User Manual Input**	**Input Validation**
postId	ID	No	No	Throw error if post not found

Output	**Type**	**Optional**	**Purpose**
deleted post	Single Post type	No	Returning information field of Post type

Front End Ruote

Name	**Ruotes**	**Component**
Discovery page(home page)	/discovery/:category?	Discover
About page	/about	About
Edit post functionality	/edit/:id	Edit
Sell functionality	/sell	Sell