Barter Bin

BARTER you will. TRADE you must!

Written By: Arnetta Knight

Changes and Updates

The Minimum Viable Product or MVP for the Barter Bin web application is a simple, yet functional platform that will allow its users to post items and services for trade, while expressing their own needs, and matching individuals based on these indicators. This can be done in large part by implementing the core features of user registration and login and the search and match function. Users will be matched based on their wants and offers, and they can search and expand their matches based on their location.

The change that will be implemented is the service layer design for Ratings and Reviews will now be moved to a new category called a “stretch feature”. It has been determined that the core functionality to support the MVP can be launched without the reviews feature. Users will still be able to successfully use the web application without the reviews function if time constraints, additional resources, and effort to implement come into play. If there are road blocks this feature can still be added later as a future release which would allow users to rate and review their experiences on the platform.

Designating ratings and reviews as a stretch feature will help to prioritize the core functionality of the application to ensure a successful launch of Barter Bin while still allowing for future improvements and enhancements.

Service Layer Design

To barter means to trade goods and services for other goods and services. Barter Bin will be a unique web application that allows its users to trade their goods and services with each other without the need of cash or other currency. For this project, I will be utilizing PHP, the PHP framework Laravel, and RESTful API to communicate with the backend of the web application. This will provide the set of tools and technologies necessary to handle requests, access data, and return responses to users. A quick overview of how these components will work together is as follows:

- PHP will provide the underlying scripting language that will power the web application. Developers primarily use PHP to write server-side code that's typically used to generate dynamic web content, database interactions, and the handling of user requests.
- Laravel is a popular PHP framework that provides a set of tools and libraries for building web applications. It's designed to have an easy learning curve and it's flexible, making it a common choice for building RESTful APIs.
- RESTful APIs are used to expose functionality to web applications, various systems, and clients over the internet. The Barter Bin application will be able to communicate with front-end clients and other web services.

These components will work together to build an efficient and scalable backend for the Barter Bin web application.

Service Layer Specifications

Overview of Functionality

- The PHP service layer will be created using the Laravel framework. The service layer contains the application logic for interacting with MySQL and handling requests from the user interface. The service layer will also include functions to perform searches, and to match users based on their wants and needs. User authentication and the ratings and review system will also be implemented in the service layer.
- The PHP service layer will interact with the database via MySQL queries to retrieve and modify data. The user interface will communicate with the service layer by sending HTTP requests to specific endpoints. The service layer will also process the requests and return the appropriate response. This will be hosted on an Amazon EC2 web server, where it will receive requests from the client to communicate with the database to retrieve and modify said data.
- The endpoints required for Barter Bin's minimal viable product or MVP will include the endpoints necessary to handle user login, item posting, searching, and matching. Rating and review posting can support the MVP, but it is not considered a "core" feature and functionality of Barter Bin. Ratings and reviews will be designated as a stretch feature. It's service layer design and flow are still unchanged.

Service Endpoints Required to Support the MVP

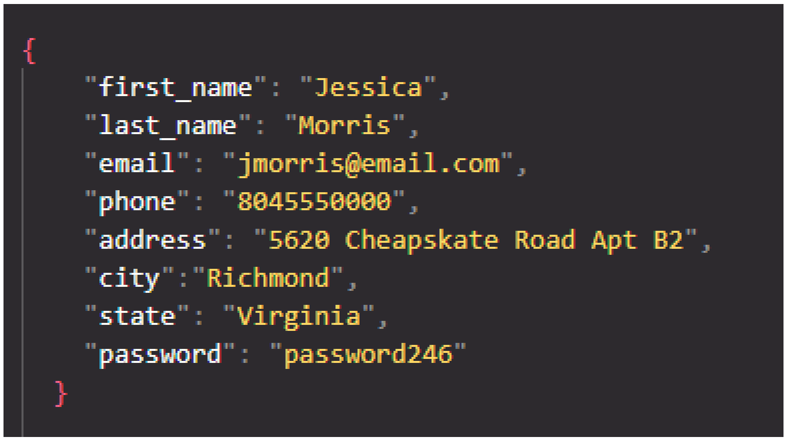
User Registration:

Method*:* POST

URL: /api/users

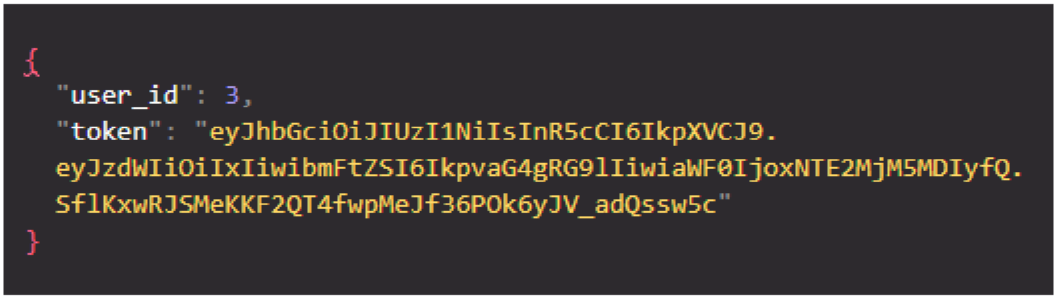
Purpose: The user registration endpoint will allow new users to the site to create an account by providing their personal information to include their first name, last name, email, phone, address, city, state, zip code, and password. When the user submits their information, the endpoint will create the new user within the database and return the success response.

Example Request:



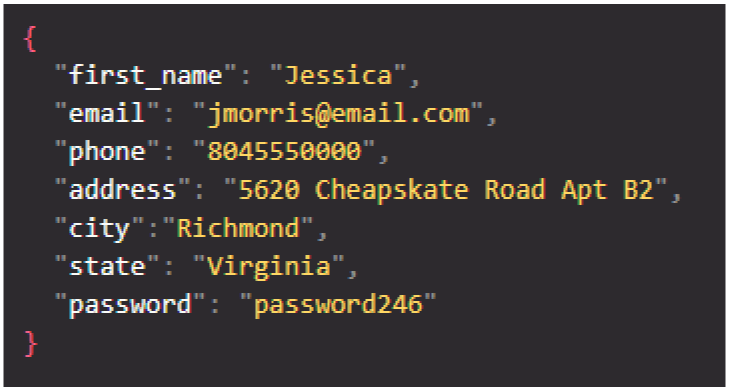
```
{
    "first_name": "Jessica",
    "last_name": "Morris",
    "email": "jmorris@email.com",
    "phone": "8045550000",
    "address": "5620 Cheapskate Road Apt B2",
    "city":"Richmond",
    "state": "Virginia",
    "password": "password246"
  }
```

Success Response: 201 CREATED



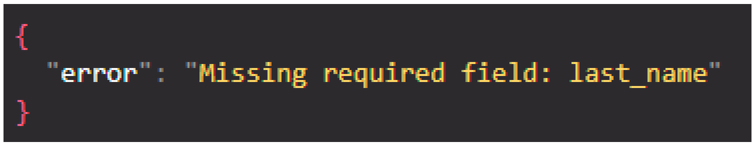
```
{
  "user_id": 3,
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
  eyJzdWIiOiIxIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.
  SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
}
```

Erroneous Request: MISSING LAST NAME



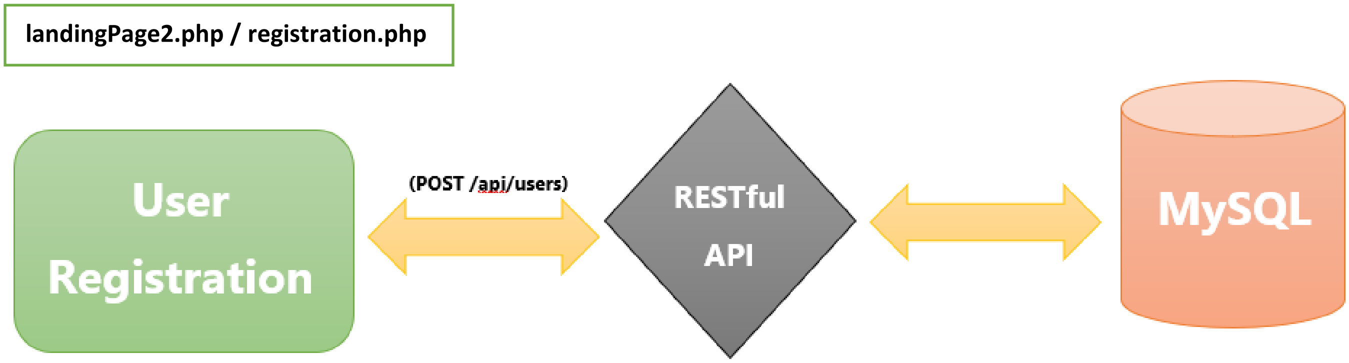
```
{
  "first_name": "Jessica",
  "email": "jmorris@email.com",
  "phone": "8045550000",
  "address": "5620 Cheapskate Road Apt B2",
  "city":"Richmond",
  "state": "Virginia",
  "password": "password246"
}
```

Error Response: 400 BAD REQUEST



```
{
  "error": "Missing required field: last_name"
}
```

Diagram:



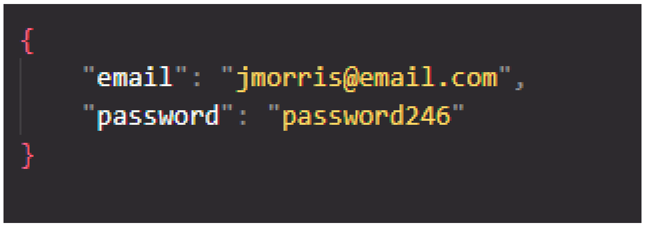
User login

Method: POST

URL: /api/login

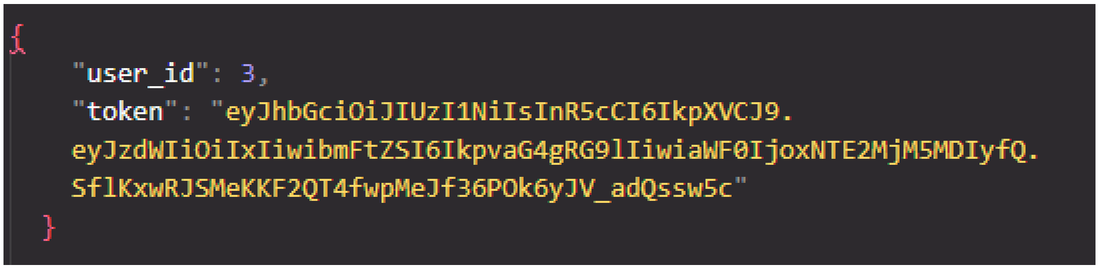
Purpose: Once users have registered for an account, this endpoint will allow them to log in to the Barter Bin web application. Users will provide their email and password. When the data is submitted, the endpoint will authenticate the users' credentials by checking them against the data stored in MySQL. If their credentials are valid, the endpoint will create a new user session and return a success response.

Example Request:



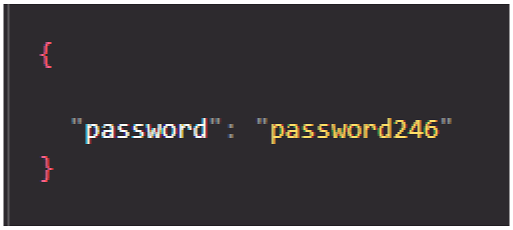
```
{
    "email": "jmorris@email.com",
    "password": "password246"
}
```

Success Response: 200 OK



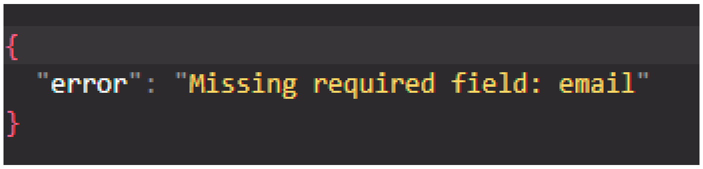
```
{
    "user_id": 3,
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
    eyJzdWIiOiIxIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.
    SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
  }
```

Erroneous Request: MISSING EMAIL



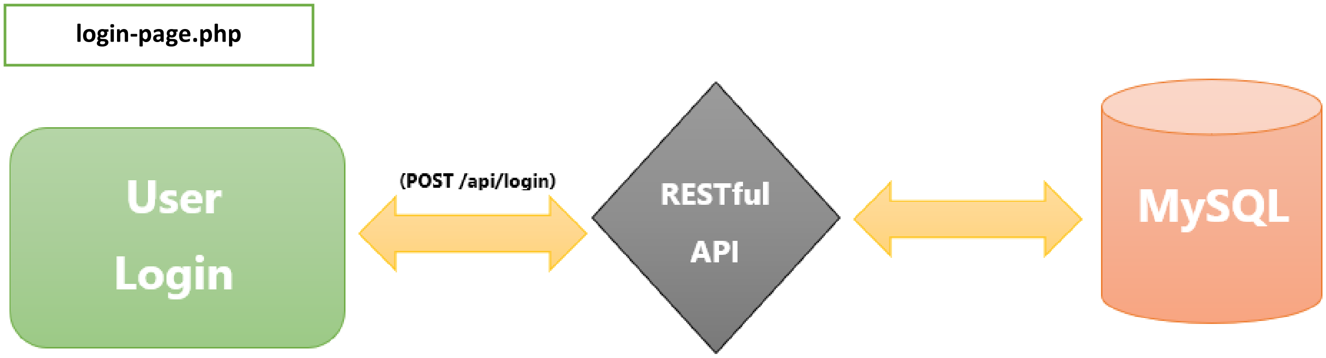
```
{
  "password": "password246"
}
```

Error Response: 400 BAD REQUEST



```
{
  "error": "Missing required field: email"
}
```

Diagram:



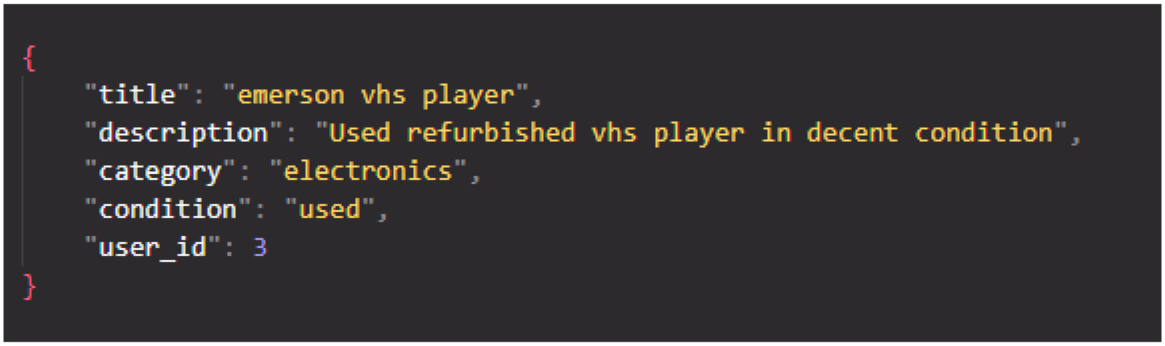
Item Posting

Method: POST

URL: /api/items

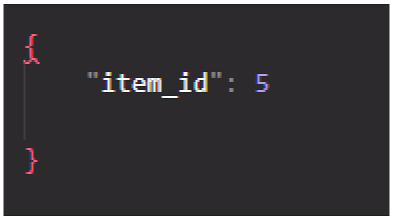
Purpose: This endpoint will allow Barter Bin users to post or "list" the items they are interested in trading. When this information is submitted by the user the endpoint will create the new item in the database and return a success response. Users will also be able to add certain details about items they have to offer such as the name of the item, description, and condition..

Example Request:



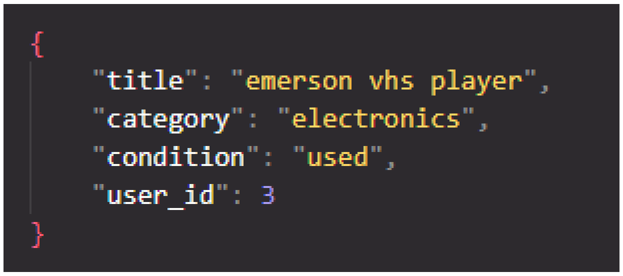
```
{
    "title": "emerson vhs player",
    "description": "Used refurbished vhs player in decent condition",
    "category": "electronics",
    "condition": "used",
    "user_id": 3
}
```

Success Response: 201 CREATED



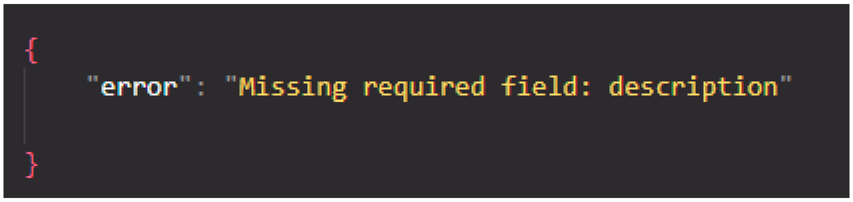
```
{
    "item_id": 5
}
```

Erroneous Request: Missing item description



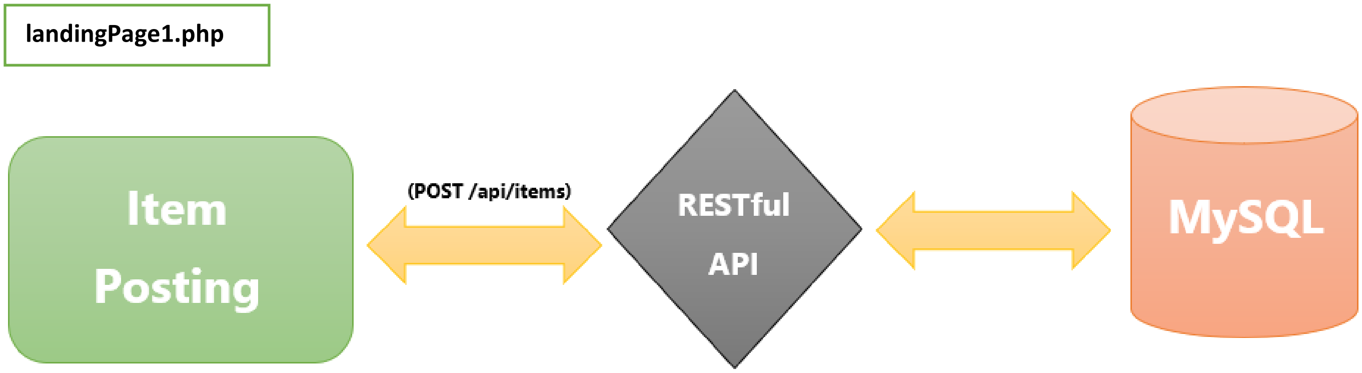
```
{
    "title": "emerson vhs player",
    "category": "electronics",
    "condition": "used",
    "user_id": 3
}
```

Error Response: 400 BAD REQUEST



```
{
    "error": "Missing required field: description"
}
```

Diagram:



Item Search:

Method: GET

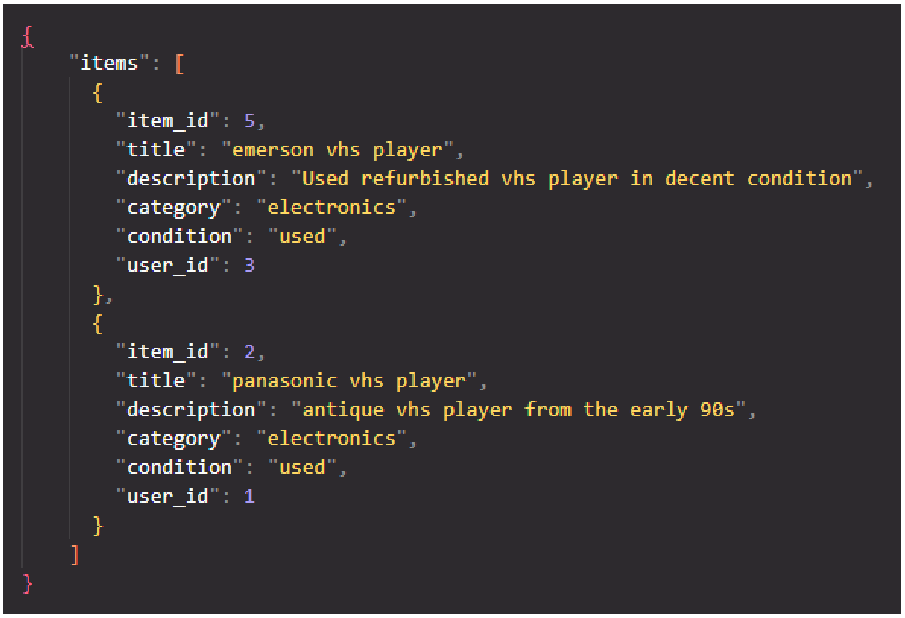
URL: /api/items?category=electronics&condition=used&title=vhs_player

Purpose: This endpoint will allow users to search for items that other users on the site have listed for trading. The search criteria will be the item that they need and their preferred city and state, or they check the "ignore location" box. This endpoint will return a list of items that closely match the criteria.

Example Request:

GET /api/items?category=electronics&condition=used&title=vhs_player

Success Response: 200 OK

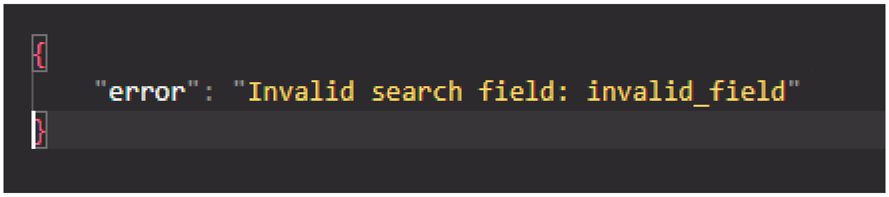


```
{
    "items": [
      {
        "item_id": 5,
        "title": "emerson vhs player",
        "description": "Used refurbished vhs player in decent condition",
        "category": "electronics",
        "condition": "used",
        "user_id": 3
      },
      {
        "item_id": 2,
        "title": "panasonic vhs player",
        "description": "antique vhs player from the early 90s",
        "category": "electronics",
        "condition": "used",
        "user_id": 1
      }
    ]
}
```

Erroneous Request:

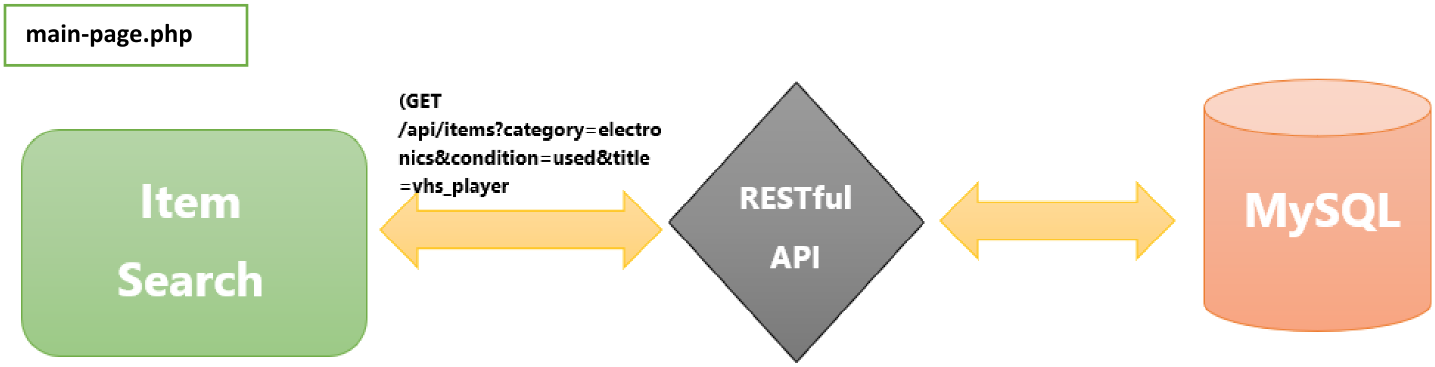
GET /api/items?invalid_field=value

Error Response: 400 BAD REQUEST



```
{
    "error": "Invalid search field: invalid_field"
}
```

Diagram:


main-page.php
Item Search
(GET /api/items?category=electronics&condition=used&title=vhs_player
RESTful API
MySQL

User Matching:

Method: GET

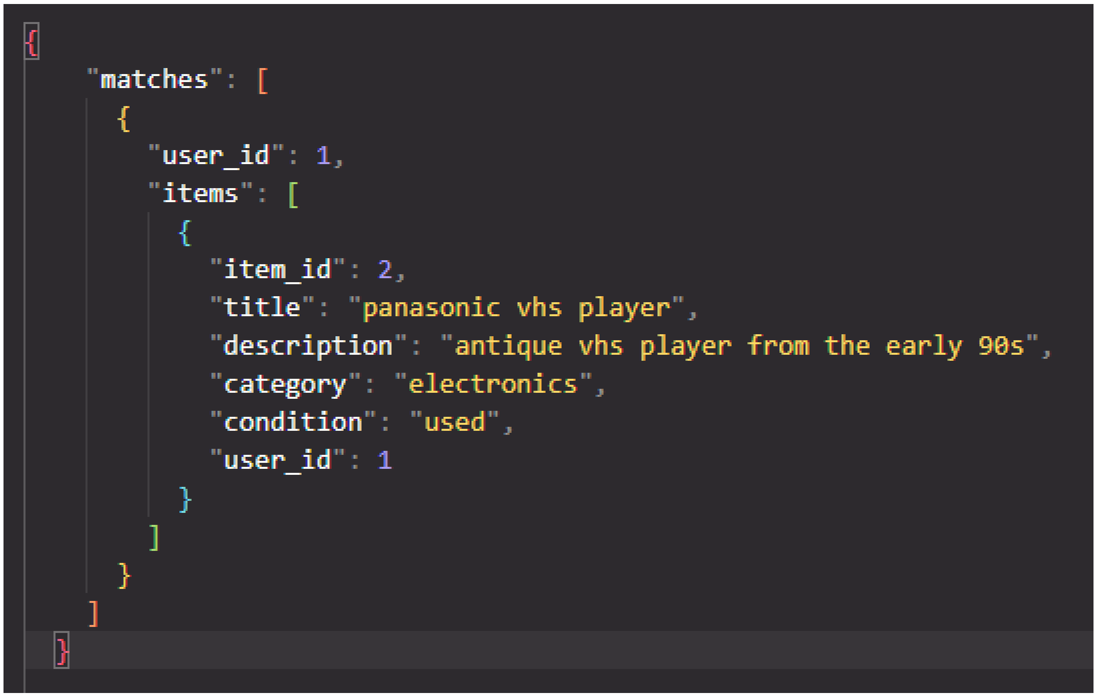
URL: /api/match?user_id=3

Purpose: This endpoint allows users to find other users who closely match the items that they need, to those users who want what they have to offer. When the item, city, and state (if applicable) criteria match it will ping both user ids within the database and return a list of users and the items they have posted for trading.

Example Request:

GET /api/match?user_id=3&category=electronics&condition=used&title=vhs_player

Success Response: 200 OK

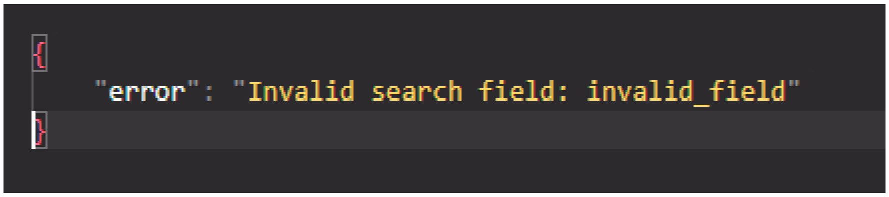


```
{
    "matches": [
      {
        "user_id": 1,
        "items": [
          {
            "item_id": 2,
            "title": "panasonic vhs player",
            "description": "antique vhs player from the early 90s",
            "category": "electronics",
            "condition": "used",
            "user_id": 1
          }
        ]
      }
    ]
}
```

Erroneous Request:

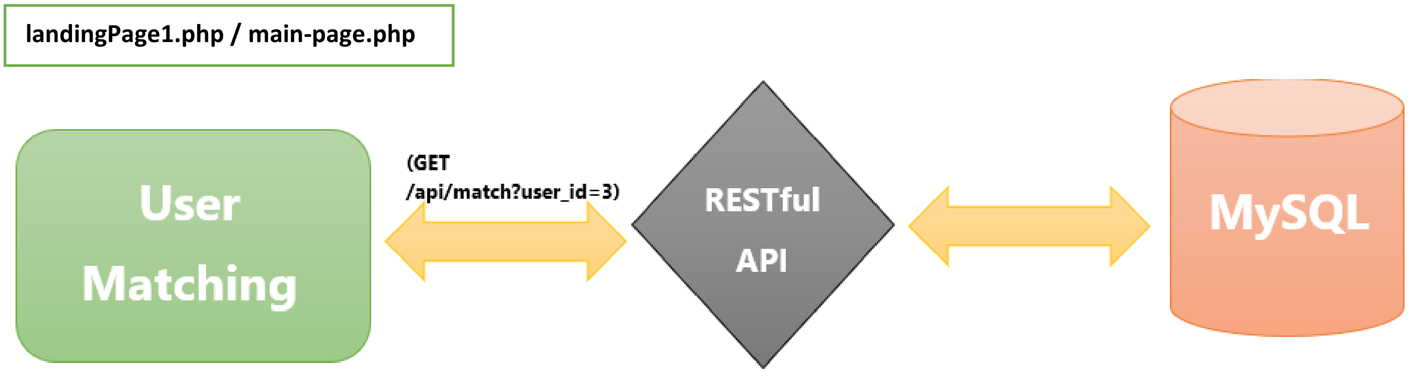
GET /api/match?invalid_field=value

Error Response: 400 BAD REQUEST



```
{
    "error": "Invalid search field: invalid_field"
}
```

Diagram:



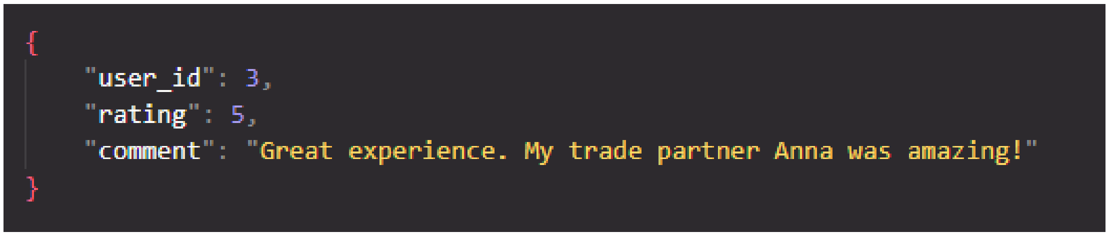
Review Posting [Stretch Feature]:

Method: POST

URL: /api/reviews

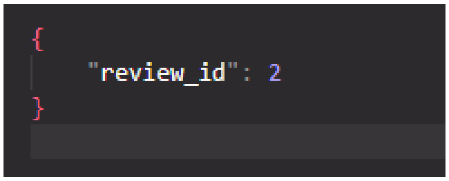
Purpose: This endpoint will allow users to rate and post reviews of their overall experience on the Barter Bin website to include their trading experiences with other users. Users can choose from a "star" rating of 1 to 5 and write a review in the provided textbox (which is optional). When the information is submitted, this will trigger the endpoint and create a new review in the database, return a success response, and the information will be displayed for the user and other site visitors/users to read.

Example Request:



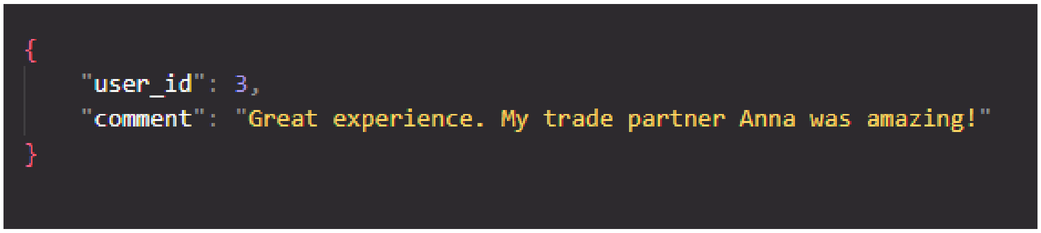
```
{
    "user_id": 3,
    "rating": 5,
    "comment": "Great experience. My trade partner Anna was amazing!"
}
```

Success Response: 201 CREATED



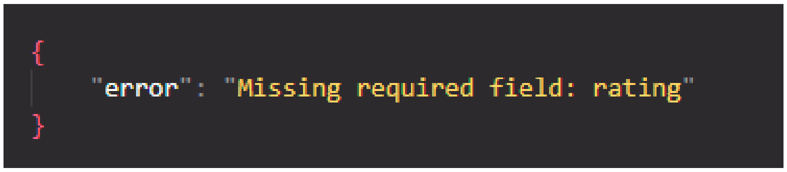
```
{
    "review_id": 2
}
```

Erroneous Request: Rating missing



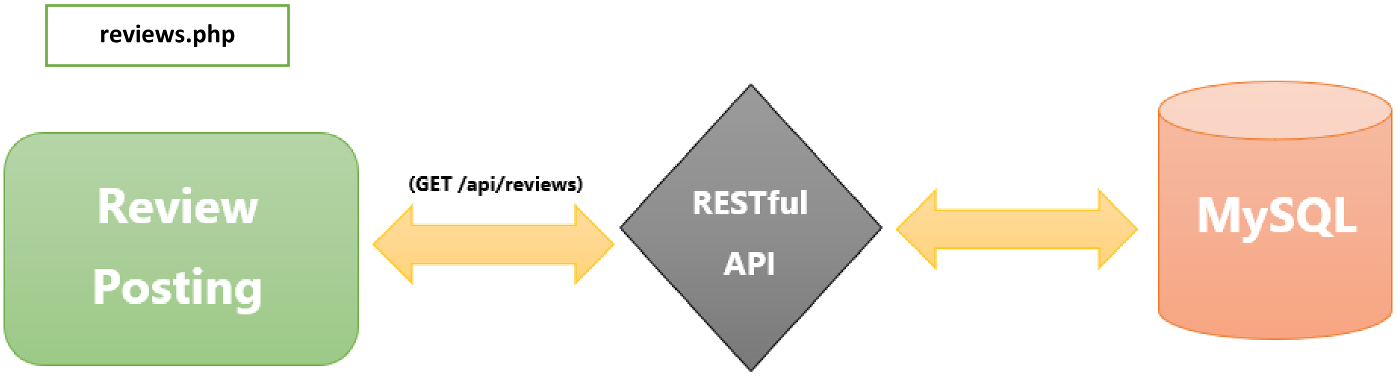
```
{
    "user_id": 3,
    "comment": "Great experience. My trade partner Anna was amazing!"
}
```

Error Response: 400 BAD REQUEST



```
{
    "error": "Missing required field: rating"
}
```

Diagram:


reviews.php
Review Posting
(GET /api/reviews)
RESTful API
MySQL