Endpoint: Get Random Pet

- **Method:** GET
- **Path:** `/random-pet`

Description: Retrieves random pet information, including profile details such as ID, name, sex, age, location, and breed. The pet information is based on a randomly generated pet ID.

Input Parameters: None

Response:

- **Success Response (200):**

```
{
   "id": "string",
   "name": "string",
   "sex": "string",
   "age": "integer",
   "location": "string",
   "breed": "string"
}
```

- **id:** Randomly generated identifier for the pet (string).
- **name:** Name of the pet (string).
- **sex:** Sex of the pet (string).
- **age:** Age of the pet (integer).
- **location:** Location of the pet (string).
- **breed:** Breed of the pet (string).

Error Responses:

- **500 Internal Server Error:** If an unexpected error occurs.

Endpoint: Add User Favorite Pet

- **Method:** POST
- **Path:** `/user/favorites/add`

Description: Adds a user’s favorite pet to the favorites database.

Input Parameters:

- **Request Body:**

```
{
    "pet_id": "string",
    "name": "string",
    "breed": "string",
    "age": "integer",
    "location": "string"
}
```

- **pet_id:** Unique identifier of the pet to add (string, required).
- **name:** Name of the pet (string, required).
- **breed:** Breed of the pet (string, required).
- **age:** Age of the pet (integer, required).
- **location:** Location of the pet (string, required).

Response:

```
{
    "favorites": [
        {
            "pet_id": "string",
            "name": "string",
            "breed": "string",
            "age": "integer",
            "location": "string"
        }
    ]
}
```

- **favorites:** List of all favorite pets after the addition.

Error Responses:

- **400 Bad Request:** If the input data is invalid.
- **500 Internal Server Error:** If an unexpected error occurs.

Endpoint: Remove User Favorite Pet

- **Method:** DELETE
- **Path:** `/user/favorites/remove`

Description: Removes a specified pet from the user's favorites database.

Input Parameters:

- **Request Body:**

```
{
  "pet_id": "string"
}
```

- **pet_id: Unique identifier of the pet to be removed (string, required).**

Response:

- **Success Response (200):**

```
{
  "favorites": [
    {
      "pet_id": "string",
      "name": "string",
      "breed": "string",
      "age": "integer",
      "location": "string"
    }
  ]
}
```

- favorites: Updated list of favorite pets after the removal.

Error Responses:

- 400 Bad Request: If the input data is invalid.
- 404 Not Found: If the specified pet is not in favorites.
- 500 Internal Server Error: If an unexpected error occurs.

Endpoint: Update User Location

- Method: PUT
- Path: `/user/location/update`

Description: Updates the user’s location information.

Input Parameters:

- Request Body:

```
{
  "Location": "string"
}
```

- Location: New location for the user (string, required).

Response:

- Success Response (200):

```
{
  "Name": "string",
  "Password": "string",
  "Location": "string",
  "Occupation": "string"
}
```

- **Name: User's name.**
- **Password: User's password.**
- **Location: Updated location of the user.**
- **Occupation: User’s occupation.**

Error Responses:

- **400 Bad Request: If the input data is invalid.**
- **500 Internal Server Error: If an unexpected error occurs.**