

REHNUMA API

API for optimizing grocery shopping by finding the lowest total cost itinerary across multiple supermarkets.

SignUp

A new user signing up as our member.

POST: Register a new user

<http://localhost:5173/SignUp>

Creates a new user account

Body: raw (json)

```
{
  "name": "<string>",
  "email": "<email>",
  "password": "<string>"
}
```

EXAMPLE:

Request: Invalid Request || Internal Server Error || User Registered Successfully.

```
var request = require('request');
var options = {
  'method': 'POST',
  'url': 'http://localhost:5173/signup',
  'headers': {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    "name": "<string>",
    "email": "<email>",
    "password": "<string>"
  })
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Response:

Body: This request does not return any response body.

Headers: This request does not return any response headers.

Just store the new user information in the database if user is registered successfully. Else ask the user to re-enter the information.

Login

A member logging into our web app.

POST: Authenticate user and return JWT token

<http://localhost:5173//Login>

Logs in a user and returns a JWT token for authentication.

Body: raw (json)

```
{  
  "email": "<email>",  
  "password": "<string>"  
}
```

EXAMPLE:

Request: Invalid email or password || Internal Server Error

```
curl --location 'http://localhost:5173//login' \  
--header 'Content-Type: application/json' \  
--header 'Accept: application/json' \  
--data '{  
  "email": "<email>",  
  "password": "<string>"  
}'
```

Response:

Body: This request does not return any response body.

Headers: This request does not return any response headers.

Ask the user to enter the credentials once again.

Request: Login successful, JWT token returned

```
var request = require('request');  
var options = {  
  'method': 'POST',  
  'url': 'http://localhost:5173//login',  
  'headers': {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json'  
  },  
  body: JSON.stringify({  
    "email": "<email>",  
    "password": "<string>"  
  })  
};  
request(options, function (error, response) {  
  if (error) throw new Error(error);  
  console.log(response.body);  
});
```

Response:

Body: json

```
{  
  "token": "<string>"  
}
```

Headers:

Content-Type: application/json

OptimizeRoute

The grocery list input by the user and outputs the optimized shopping guide.

POST: Get the optimized shopping itinerary

<http://localhost:5173//OptimizeRoute>

Receives a user's grocery list and returns the best route considering supermarket prices and fuel costs. Calls the backend Python function *find_best_itinerary()* for optimization.

Authorization Bearer Token

Token {{bearerToken}}

Body: raw (json)

```
{
  "userLocation": {
    "latitude": "<float>",
    "longitude": "<float>"
  },
  "groceryList": [
    "<string>",
    "<string>"
  ]
}
```

EXAMPLE:

Request: Internal Server Error || Invalid Request Format || Unauthorized User

```
var request = require('request');
var options = {
  'method': 'POST',
  'url': 'http://localhost:5173//optimize-route',
  'headers': {
    'Content-Type': 'application/json',
    'Authorization': 'Bearer <token>'
  },
  body: JSON.stringify({
    "userLocation": {
      "latitude": "<float>",
      "longitude": "<float>"
    },
    "groceryList": [
      "<string>",
      "<string>"
    ]
  })
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Response:

Body: This request does not return any response body.

Headers: This request does not return any response headers.

Request: Optimized Shopping Itinerary Generated

```
var request = require('request');
var options = {
  'method': 'POST',
  'url': 'http://localhost:5173//optimize-route',
  'headers': {
    'Content-Type': 'application/json',
    'Accept': 'application/json',
    'Authorization': 'Bearer <token>'
  },
  body: JSON.stringify({
    "userLocation": {
      "latitude": "<float>",
      "longitude": "<float>"
    },
    "groceryList": [
      "<string>",
      "<string>"
    ]
  })
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Response:

Body: json

```
{
  "itinerary": [
    {
      "supermarket": "<string>",
      "address": "<string>",
      "items": [
        "<string>",
        "<string>"
      ],
      "totalCost": "<float>"
    },
    {
      "supermarket": "<string>",
      "address": "<string>",
      "items": [
        "<string>",
        "<string>"
      ],
      "totalCost": "<float>"
    }
  ]
}
```

Headers:

Content-Type: application/json

History

User can view their previously generated optimized shopping plans.

GET: Get user's past shopping itineraries

<http://localhost:5173//History>

Fetches the previously generated itineraries for the logged-in user.

Authorization Bearer Token

Token {{bearerToken}}

EXAMPLE:

Request: Internal Server Error || Unauthorized User

```
var request = require('request');
var options = {
  'method': 'GET',
  'url': 'http://localhost:5173//history',
  'headers': {
    'Authorization': 'Bearer <token>'
  }
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Response:

Body: This request does not return any response body.

Headers: This request does not return any response headers.

Display error message to the user.

Request: List of past itineraries retrieved successfully.

```
var request = require('request');
var options = {
  'method': 'GET',
  'url': 'http://localhost:5173//history',
  'headers': {
    'Accept': 'application/json',
    'Authorization': 'Bearer <token>'
  }
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

Response:

Body: json

```
[
  {
    "itinerary": [
      {
        "supermarket": "<string>",
        "address": "<string>",
        "items": [
          "<string>",
          "<string>"
        ],
        "totalCost": "<float>"
      },
      {
        "supermarket": "<string>",
        "address": "<string>",
        "items": [
          "<string>",
          "<string>"
        ],
        "totalCost": "<float>"
      }
    ]
  },
  {
    "itinerary": [
      {
        "supermarket": "<string>",
        "address": "<string>",
        "items": [
          "<string>",
          "<string>"
        ],
        "totalCost": "<float>"
      },
      {
        "supermarket": "<string>",
        "address": "<string>",
        "items": [
          "<string>",
          "<string>"
        ],
        "totalCost": "<float>"
      }
    ]
  }
]
```

Headers:

Content-Type: application/json