HealthyEats 2.0 — Complete Postman Test Suite

This collection covers the full user flow from registration through AI meal generation and history retrieval. Replace `<BASE_URL>`, `<JWT_TOKEN>`, `<USER_ID>`, `<PROFILE_ID>`, `<PREF_ID>` as needed.

---

1. Register New User

Request

```
POST <BASE_URL>/api/user/register
Headers:
  Content-Type: application/json

Body:
{
  "name": "Test User",
  "email": "test.user@example.com",
  "password": "Password123!"
}
```

Expected Response (201)

```
{
  "message": "User registered successfully",
  "user": {
    "_id": "<USER_ID>",
    "name": "Test User",
    "email": "test.user@example.com",
    "createdAt": "2025-06-20T00:00:00.000Z"
  }
}
```

---

2. Login & Retrieve JWT

Request

```
POST <BASE_URL>/api/user/login
Headers:
  Content-Type: application/json

Body:
{
  "email": "test.user@example.com",
  "password": "Password123!"
}
```

Expected Response (200)

```
{
  "token": "<JWT_TOKEN>",
  "user": {
    "_id": "<USER_ID>",
    "name": "Test User",
    "email": "test.user@example.com"
  }
}
```

Save `<JWT_TOKEN>` for all subsequent calls under `Authorization: Bearer <JWT_TOKEN>`.

---

3. Create Onboarding Profile

Request

```
POST <BASE_URL>/api/user-profile
Headers:
  Content-Type: application/json
  Authorization: Bearer <JWT_TOKEN>

Body:
{
  "age": 30,
  "gender": "female",
  "heightFeet": 5,
  "heightInches": 7,
  "weight": 65
}
```

Expected Response (201)

```
{
  "_id": "<PROFILE_ID>",
  "userId": "<USER_ID>",
  "age": 30,
  "gender": "female",
  "heightFeet": 5,
  "heightInches": 7,
  "weight": 65,
  "createdAt": "2025-06-20T00:05:00.000Z",
  "updatedAt": "2025-06-20T00:05:00.000Z"
}
```

4. Fetch FoodAI Question Definitions

Request

```
GET <BASE_URL>/api/foodAI/questions
Headers:
  Authorization: Bearer <JWT_TOKEN>
```

Expected Response (200)

```
[
  { "key":"foodTypes",       "label":"Preferred food type",
"type":"select",      "options":["vegetarian","vegan","low-carb","high-
protein","any"] },
  { "key":"allergies",       "label":"Allergies or foods to
avoid","type":"multiselect","options":
["nuts","shellfish","dairy","gluten","soy","none"] },
  { "key":"medicalConditions","label":"Health conditions",
"type":"multiselect","options":
["heart","kidney","liver","diabetes","none"] },
  { "key":"diabeticRange",   "label":"If diabetic, specify blood sugar
range","type":"text","dependsOn":"medicalConditions" },
  { "key":"pregnancyStatus","label":"Pregnancy status",
"type":"toggle",      "options":["yes","no"] },
  { "key":"budget",          "label":"Budget per meal (BDT)",
"type":"number" }
]
```

5. Save FoodAI Questionnaire Answers

Request

```
POST <BASE_URL>/api/foodPreferences
Headers:
  Content-Type: application/json
  Authorization: Bearer <JWT_TOKEN>

Body:
{
  "userId": "<USER_ID>",
  "foodTypes": ["vegetarian"],
  "allergies": ["nuts"],
  "medicalConditions": ["diabetes"],
  "diabeticRange": "120-140",
  "pregnancyStatus": true,
  "budget": 300
}
```

Expected Response (200)

```
{
  "_id": "<PREF_ID>",
  "userId": "<USER_ID>",
  "foodTypes": ["vegetarian"],
  "allergies": ["nuts"],
  "medicalConditions": ["diabetes"],
  "diabeticRange": "120-140",
  "pregnancyStatus": true,
  "budget": 300,
  "createdAt": "2025-06-20T00:06:00.000Z",
  "updatedAt": "2025-06-20T00:06:00.000Z"
}
```

---

6. Generate & Save AI Meal Plan

Request

```
POST <BASE_URL>/api/foodAI/generate
Headers:
  Content-Type: application/json
  Authorization: Bearer <JWT_TOKEN>

Body:
{
  "userId": "<USER_ID>"
}
```

Expected Response (200)

```
{
  "message": "Meal plan generated and saved!",
  "mealPlan": "=== Breakfast ===\nFoods: Oats (½ cup), Vegetable Curry...
\nFruits: ½ cup sliced papaya...\n...\n=== Lunch ===\nFoods: Rice (1 cup)...
\n..."
}
```

7. (Optional) Fetch Meal History

Request

```
GET <BASE_URL>/api/meals/history/<USER_ID>
Headers:
  Authorization: Bearer <JWT_TOKEN>
```

Expected Response (200)

```
[
  {
    "_id": "...",
    "date": "2025-06-19T20:40:43.763Z",
    "mealType": "fullDay",
    "menu": ["=== Breakfast ===\n..."],
    "aiSource": "Gemini v2 Render",
    "userId": "<USER_ID>",
    "createdAt": "2025-06-19T20:40:43.772Z"
  },
  // …more entries
]
```

Notes:

- Ensure your server is running at `<BASE_URL>` (e.g., `http://localhost:3000`).
- Use the same `<JWT_TOKEN>` for all secured calls.
- Replace placeholder IDs with actual values obtained during testing.