CS 340 Project Step 6 for My Kidney Nutrition Tracker Application

Project Group 70-Charles Cal and Deanna Denny

URL: http://flip3.engr.oregonstate.edu:3331/

(have to be connected to VPN)

Executive Summary

In the beginning of our project we had more feedback and issues to resolve as we learned more about databases from the class and as we gained more knowledge and feedback too from our peers/grader. We determined and filtered what was applicable to add within our project. We even overcame odd glitches in Flask and the Flip servers that caused our database to time out constantly.

Throughout this project we were heavily influenced by peer reviews, instructor feedback, attendance in office hours seeking more advice, and collaboration within our group. Early on, feedback focused on database design. Based on instructor feedback we eliminated a 1:M relationship between Dialysis_Forms and Patients, as this relationship was redundant with the 1:M relationship between Patients and Lab_Results. To ensure we maintained naming consistency, we changed the names of all attributes tracking units to their singular form. Also for naming consistency, we renamed our “Food” table to “Foods.” We identified that we needed to include both an ER diagram and a schema to convey both detailed and generalized models for our database. We achieved this by re-creating our ER diagram with a simplified model, focusing only on relationships between entities, primary, and foreign keys.

We then updated our schema based on normalization principles learned in the “Module 5: Normalization Steps” exploration. We learned from this module that our database was not normalized as we thought from online research we conducted previously so we eliminated further redundant data such as units for each lab value and food nutrient units.

We received several pieces of feedback that suggested expanding the scope of our database. We tried to implement these suggestions, but ultimately opted for a simpler model to practice the principles we learned in the class explorations.

We deployed our application on the flip servers using gunicorn and the class MySQL database. At this point, we struggled to troubleshoot an issue that was causing our database connection to timeout. Based on peer and ULA suggestions, we rectified this issue after identifying that it was ultimately caused by multiple database connections set up within each route. We refactored our code for readability and to add citations and improved our UI front-end appearance. We finished all the required CRUD operations and improved their functionality. We used the Lab Results relationships with Patients and with Dialysis_Forms as our nullable relationships, and ensured our UI supported adding null values for each of these to the Lab_Results table.

Overall, we took a considerable amount of time to reflect on the feedback that was given and implemented what was most applicable in the creation of our final project.

Project Outline and Database Outline:

Overview

Kidney disease impacts around 37 million people within the United States and is one of the leading causes of death. Health insurances pay an estimated $87.2 billion to treat chronic kidney disease and around $37.3 billion to treat those who have end-stage renal disease.[1] Around 81% of kidney disease patients own a cell phone that has the ability to connect to mobile applications.[2] More generations are tech savvy and becoming more reliant on telehealth/web development technologies. There are 786,000 patients with end-stage renal disease with appropriately 550,000 on dialysis within the United States.[3] We would anticipate that our application would start off with a goal of reaching 100,000 patients but a long-term goal of reaching the 81% of dialysis population with mobile phones that have application access.

The diet for individuals who have kidney disease is one of the most complicated diets to follow which include potentially restricting phosphorus, sodium, and potassium. There are currently more than 165,000 health-related and diet applications where only one-third of these actually focus on chronic diseases. Most of the current kidney-related applications information that is provided are not accurate or evidence-based. Many of them such as Kidney APPetite, Kidney Diet and Pocket Dietitian have been discontinued with newer operating systems.[4]

Our database driven website will provide a niche that is not currently being represented in today's market for dialysis. Especially bringing in renal dietitian skills paired with development; there is currently nothing that exists with this experience mixture but needs to in order to provide accurate and evidence-based information to those that struggle with their diet and require this information to be accessible to them. In this database, we will provide our Patient (on dialysis) with a tracking system for their Foods consumed which will allow the Patient to track specific nutritional content that is renal-focused to help them achieve their health goals/Lab Results that are within metric benchmarks. These Lab Results are influenced by what Foods the Patient decides to consume and impacted by the Dialysis Form they are on as well. This database will provide the Patient with the resources ultimately to achieve better outcomes while tracking certain attributes that influence their abilities to achieve these goals.

Outside the scope of this project, we would look at including additional details, such as prohibited foods for transplant patients, or a limited record of what drugs

patients may be taking. We also see this database as forming a structure that could be replicated to address other dietary-related health disorders.

Database Outline

Entities & Attributes:

Patients: Records the details of the patients. Focusing on patients with end-stage renal disease that are on dialysis.

- patient_id: INT (11), auto_increment, unique, not NULL, PK
- last_name: VARCHAR(128), not NULL
- first_name: INT(128), not NULL
- age: INT (11), not NULL
- gender: VARCHAR(20), default NULL
- height: INT(11), not NULL
- weight: INT(11), not NULL
- Using customary units for height and weight assumed to be reported as weight in pounds and height in inches.
- relationship: M:N relationship with Foods implemented with Patients_patient_id and Foods_food_id as a FK inside of Patients_Foods. Along with a timestamp of patient_food_time DATETIME().
- relationship: 1:M relationship with Lab_Results implemented with Patients_patient_id as a FK inside of Lab_Results.

Foods: records the foods the patients consume focusing on minerals/calories that need to be watched in the renal diet. These main minerals include phosphorus, potassium, and sodium. Along with focusing on calories consumed too. Recommended units for potassium, sodium, and phosphorus are mg, and the recommended unit for calories is kcals. Amount is a the number of grams in a serving size.

- food_id: INT (10), auto_increment, unique, not NULL, PK
- food_name: VARCHAR(128), not NULL
- phosphorus_content: INT(11), default NULL
- sodium_content: INT(11), default NULL
- calories: INT(11), default NULL
- potassium_content: INT(11), default NULL
- amount: INT(11), default NULL

- Using Metric system's standardized units: potassium/phosphorus/sodium nutrient content would be in mg, and calories would be in kcals. Food portions/amounts would be reported in grams.
- relationship: M:N relationship with Patients implemented with Foods_food_id and Patients_patient_id as a FK inside of Patients_Foods. Along with a timestamp of patient_food_time DATETIME().

Lab_Results: records the lab results of the kidney patients that are nutritionally relevant such as phosphorus, potassium and sodium levels. It is also important to monitor adequacy or also known as Kt/v which is a metric that shows how well the patient is dialyzing which is heavily influenced by the type of dialysis they are on since different forms of dialysis have different adequacy metrics to meet. Recommended units for phosphorus is mg/dL, and for sodium and potassium is mEq/L.

- lab_id: INT(11), auto_increment, unique, not NULL, PK
- phosphorus_lab: FLOAT
- potassium_lab: FLOAT, default NULL
- sodium_lab: INT(11), default NULL
- dialysis_adequacy_lab: FLOAT, default NULL
- lab_results_time: DATETIME(), default NULL
- Utilizes the Conventional Unit system for laboratory values which is adapting the standardized units used within the United States for these lab values: sodium mEq/L, potassium mEq/L, and phosphorus mg/dL.
- relationship: M:1 relationship with Patients implemented with Patients_patient_id as a FK inside of Lab_Results. This is a nullable relationship, as this entry is optional.
- relationship: M:1 relationship with Dialysis_Forms implemented with Dialysis_Forms_dialysis_id as a FK inside of Lab Results. This is a nullable relationship, as this entry is optional.

Dialysis_Forms: records the type of dialysis the kidney patient is on; common types of dialysis include in-center, home-hemo, and peritoneal dialysis. The form of dialysis can affect the desired lab results, and different forms of dialysis have different desired adequacy metrics.

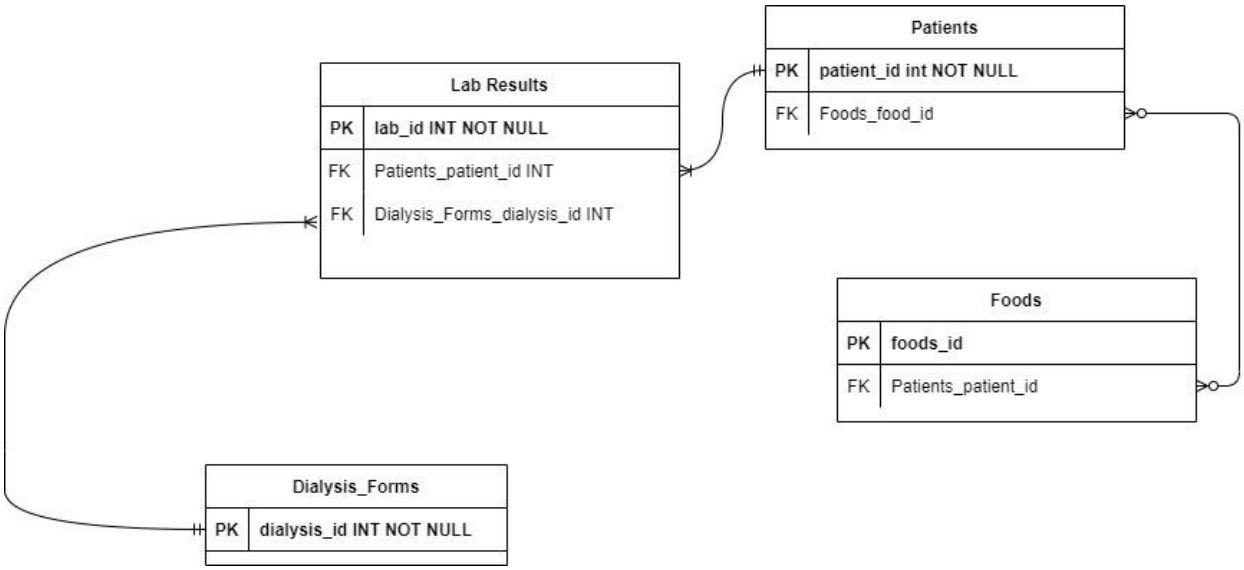
- dialysis_id: INT(10), auto_increment, unique, not NULL, PK
- name: VARCHAR(128), not NULL
- location_type: VARCHAR(128), not NULL
- adequacy_standard: FLOAT, not NULL
- relationship: 1:M relationship with Lab_Results implemented with Dialysis_Forms_dialysis_id as a FK inside of Lab_Results.

Intersection Table:

Patients_Food: Intersection table between Patients and Foods. This connects the patients to food that they have eaten, and adds a time characteristic for when the patient ate the food.

- food_patient_id: INT, PK
- Foods_food_id: INT (10), FK ON DELETE CASCADE
- Patients_patient_id: INT (11), FK ON DELETE CASCADE
- patient_food_time: DATETIME(), not NULL

Entity-Relationship Diagram:



---

[1] Centers for Disease Control and Prevention. (2022, February 28). *Chronic Kidney Disease Basics*. Centers for Disease Control and Prevention. Retrieved April 3, 2022, from https://www.cdc.gov/kidneydisease/basics.html#:~:text=15%25%20of%20US%20adults%20are,is%20about%2037%20million%20people.

[2] Singh, K. (2021, January 7). *Mobile Health in dialysis: The best engagement medium is the one that's with patients*. American Society of Nephrology. Retrieved April 14, 2022, from https://cjasn.asnjournals.org/content/16/1/12

[3] U.S. Department of Health and Human Services. (n.d.). *Kidney Disease Statistics for the United States*. National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved April 14, 2022, from https://www.niddk.nih.gov/health-information/health-statistics/kidney-disease

[4] *Renal diet apps: Which one should I use?* Kidney Diet Tips. (2018, July 30). Retrieved April 3, 2022, from https://blogs.davita.com/kidney-diet-tips/renal-diet-apps-which-one-should-i-use/

Schema:

To normalize our schema we first created tables that we compiled that would fit our report structure in excel. We eliminated applicable redundant data to make it a first normal form (1NF). We then identified the primary keys within our tables and dependencies. Then we eliminated partial dependencies so that it fits the second normal form (2NF). We reassigned/omitted applicable dependent attributes. We then ensured it was in the third normal form (3NF) by making sure we resolved all transitive dependencies. We followed the normalization steps that were in our module 5 exploration on normalization steps.[1]

Part of these normalization steps, we decided to utilize the Conventional Unit system for laboratory values which is adapting the standardized units used within the United States for these lab values: sodium mEq/L, potassium mEq/L, and phosphorus mg/dL.[2] Since they are based on the Conventional Unit system there is no need to explicitly label each of them in the database on the Lab_Results table. For food measurements we used the Metric system's standardized units: potassium/phosphorus/sodium nutrient content would be in mg, and calories would be in kcals within the Foods table. Food portions/amounts would be reported in grams on the Foods table.[3] We are using Customary units for weight and height in the Patients table which reports weight as pounds and height in inches.[4] We decided since we are using the typical standards that are reported within the dialysis community and in the United States that we decided to leave that information out of the database we designed.

By standardizing units with commonly used unit systems within the United States we were able to successfully decrease the amount of duplication that comes from our previous database and more so fitting the expected guidelines within a 3NF. It also eliminated the partial dependencies as well. See on next page the detailed schema model.

---

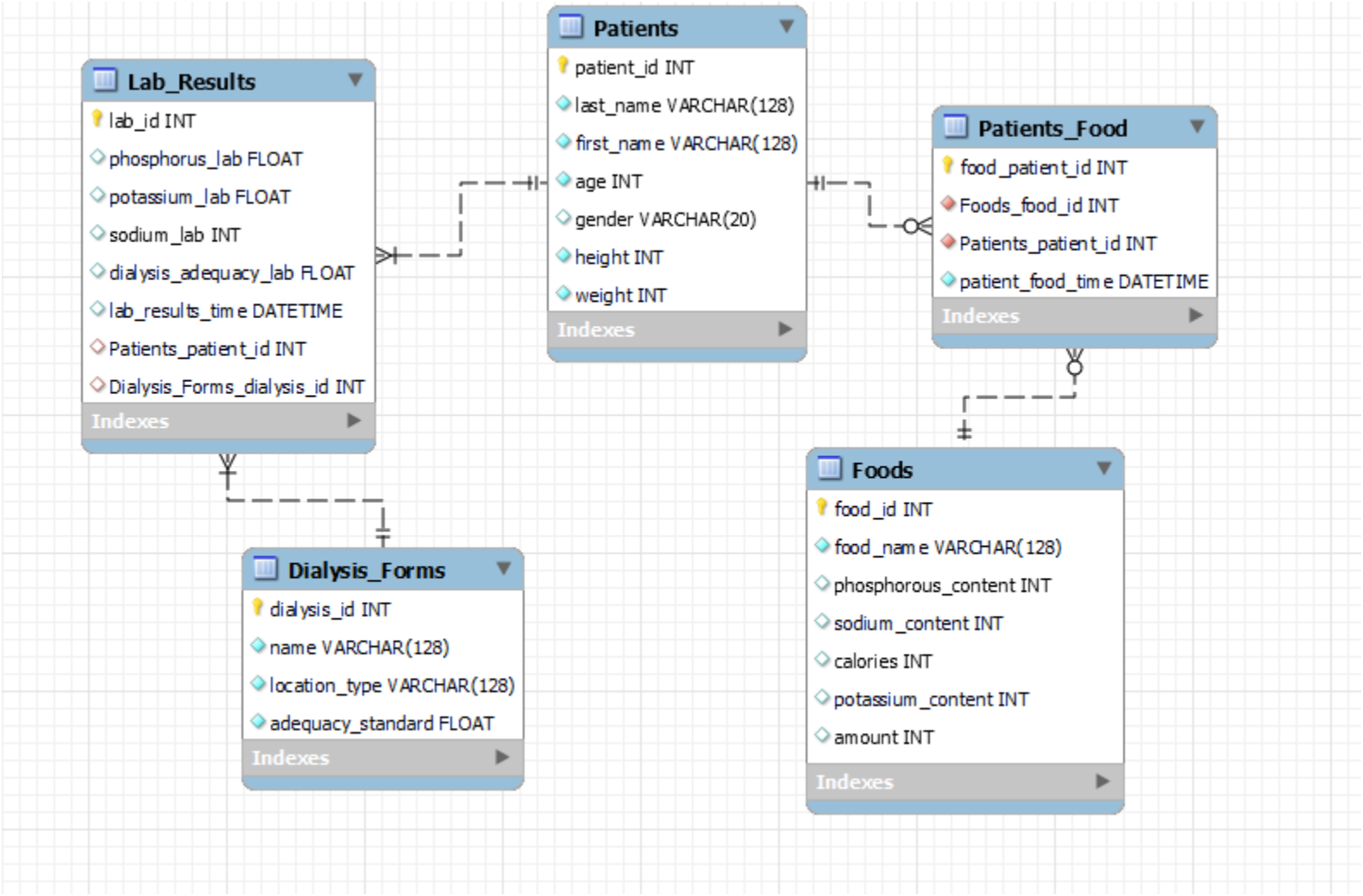
1 https://canvas.oregonstate.edu/courses/1870053/pages/exploration-normalization-steps?module_item_id=22036024

2 *Conventional units – International Units*. GlobalRPH. (n.d.). Retrieved April 26, 2022, from https://globalrph.com/medical/conventional-units-international-units/

3 *Your guide to the New Food Label*. National Kidney Foundation. (2022, January 13). Retrieved April 26, 2022, from https://www.kidney.org/atoz/content/foodlabel

4 *What are customary units? - definition, facts and examples*. What are Customary Units? - Definition, Facts and Examples. (n.d.). Retrieved April 27, 2022, from https://www.splashlearn.com/math-vocabulary/measurements/customary-units

Schema Model:



Example Data:

All sample data is based on fictional patients. Lab standards, dialysis forms, and food data are referenced from sources below. Refer as well to our SQL file containing our Data Definition Queries (DDL) for more information.

Patients Table:

		Patients				
patient_id	last_name	first_name	age	gender	height	weight
1	Smith	Arlene	55	F	64	145
2	Rogers	Christopher	63	M	72	180
3	Harrison	Kayla	68	F	65	125
4	Jackson	Henry	74	M	75	200
5	Wonders	Brenda	91	F	60	92

Patient information is fictional
patient_id = auto generating
Height in inches and weight in lbs considered Customary standardized units.

Lab_Results Table:[5]

	Lab_Results						
lab_id	phosphorus_lab	potassium_lab	sodium_lab	dialysis_adequacy_lab	lab_results_time	Patients_patient_id	Dialysis_Forms_dialysis_id
1	3.5	3.4	135	1.2	2022-05-07 23:22:05	3	1
2	5.5	3	142	1.7	2022-05-08 18:36:10	2	2
3	6.5	2.8	146	1.1	2022-05-01 20:20:06	4	1

5 *Understanding your lab work*. DaVita. (n.d.). Retrieved April 25, 2022, from https://www.davita.com/education/kidney-disease/symptoms/understanding-your-lab-work

4	10.5	6.6	144	0.6	2022-05-07 18:01:55	5	1
5	7.2	4.5	134	2.2	2022-05-11 10:19:25	1	2

lab_id, Patients_patient_id, Dialysis_Forms_dialysis_id = auto generating
Using Conventional standardized units for lab values: potassium and sodium measured in mEq/L and phosphorus in mg/dL. Adequacy is measured as a benchmark metric with hemodialysis achieving 1.2 and peritoneal dialysis is achieving 1.7.

Dialysis_Forms Table:[6] [7] [8]

		Dialysis_Forms	
dialysis_id	name	location_type	adequacy_standard
1	hemodialysis FMC	incenter	1.2
2	peritoneal Baxter	home	1.7

Dialysis_id = auto generating
Adequacy is measured as a benchmark metric with hemodialysis patients achieving 1.2 and peritoneal dialysis 1.7.

---

6 *What is dialysis?* National Kidney Foundation. (2022, February 4). Retrieved April 25, 2022, from https://www.kidney.org/atoz/content/dialysisinfo

7 U.S. Department of Health and Human Services. (n.d.). *Peritoneal dialysis: Dose & Adequacy*. National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved April 25, 2022, from https://www.niddk.nih.gov/health-information/professionals/clinical-tools-patient-management/kidney-disease/identify-manage-patients/manage-ckd/peritoneal-dialysis-dose-adequacy

8 U.S. Department of Health and Human Services. (n.d.). *Hemodialysis: Dose & adequacy*. National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved April 25, 2022, from https://www.niddk.nih.gov/health-information/professionals/clinical-tools-patient-management/kidney-disease/identify-manage-patients/manage-ckd/hemodialysis-dose-adequacy#:~:text=The%20two%20methods%20generally%20used,blood%20flow%20through%20the%20dialyzer.

Foods_Table:[9]

		Foods				
food_id	name	amount	phosphorus_content	sodium_content	calories	potassium_content
1	Milk, whole	128	251	94.6	152	374
2	Beef, loin, top loin steak	284	585	128	423	801
3	Chicken, breast	174	419	81.8	275	597
4	Yogurt, Greek, nonfat	156	212	56.2	92	220
5	Kale	100	55	53	43	348

Food_id = auto generating
Units of measurement are based on the US metric system: phosphorus/sodium/potassium in milligrams(mg), calories as kcals, and amount/portion in grams.

Patient_Food Table:

		Patients_Food	
food_patient_id	Foods_food_id	Patients_patient_id	patient_food_time
1	5	1	2022-05-10 15:40:11
2	4	2	2022-05-20 18:32:04
3	3	3	2022-05-15 12:08:12
4	2	4	2022-05-11 15:07:55
5	1	5	2022-05-16 10:22:28

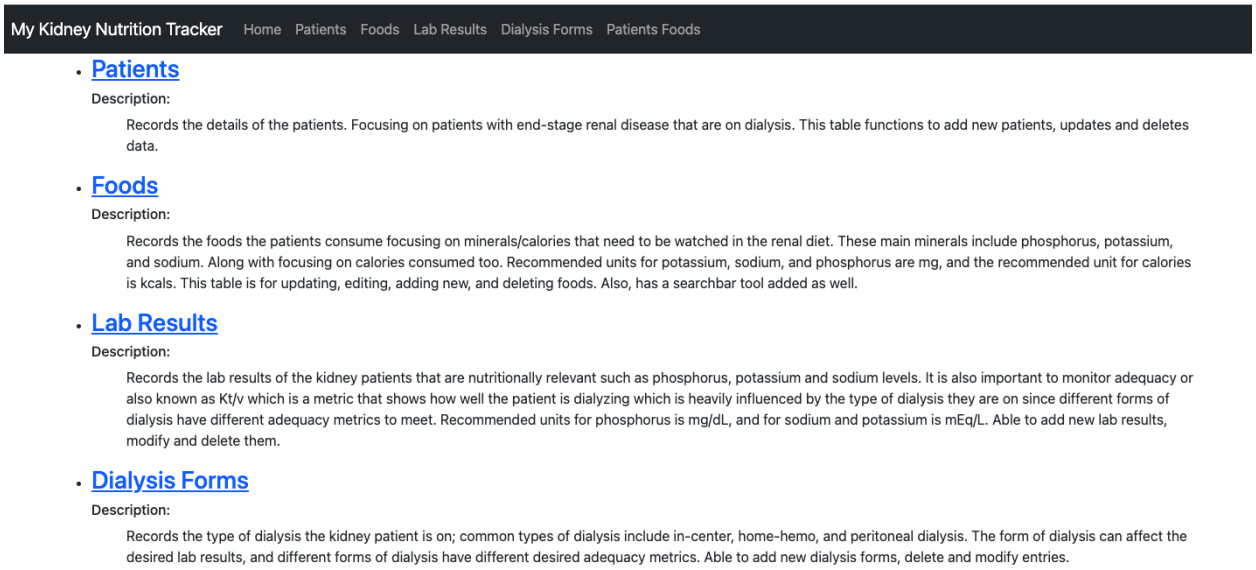
9 *Fooddata Central Search Results*. FoodData Central. (n.d.). Retrieved April 25, 2022, from https://fdc.nal.usda.gov/fdc-app.html#/

Foods_food_id and Patients_patient_id = FKs referencing Foods and Patients tables.

UI Screenshots:

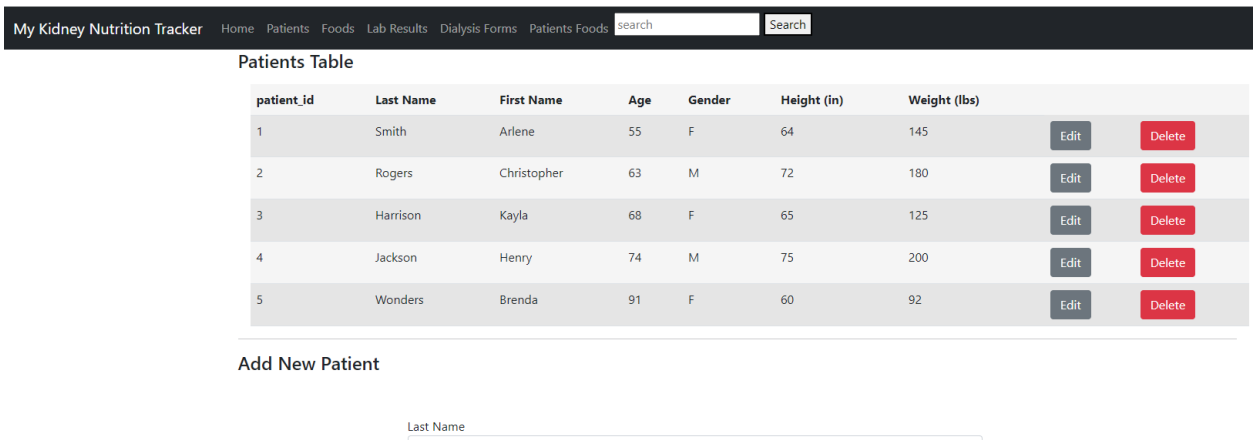
Home Page:

READ/BROWSE/DISPLAY Home page

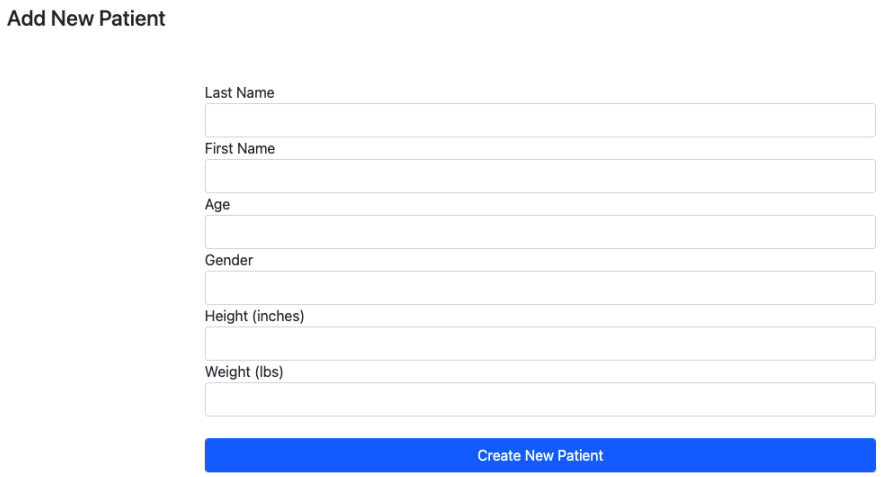


Patients Page:

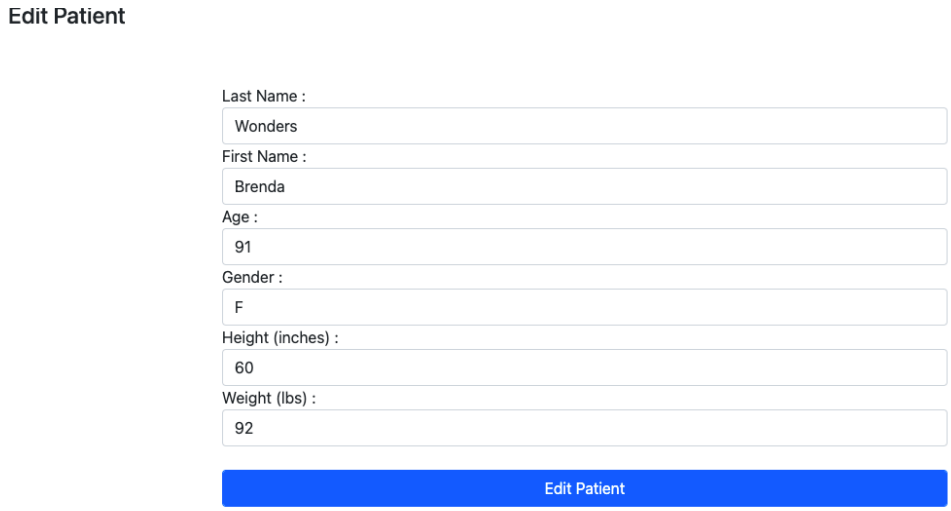
READ/BROWSE/DISPLAY Patients page



CREATE/INSERT/ADD New Patients page



UPDATE Patients page:



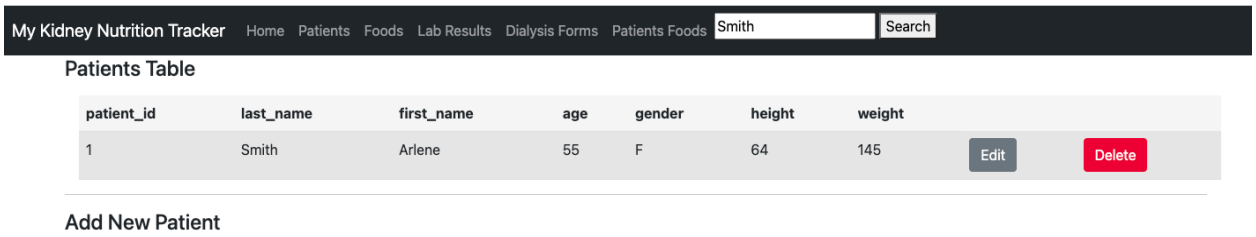
DELETE Patients page

Remove Patient

Are you sure you want to remove Patient ID: 7 ?

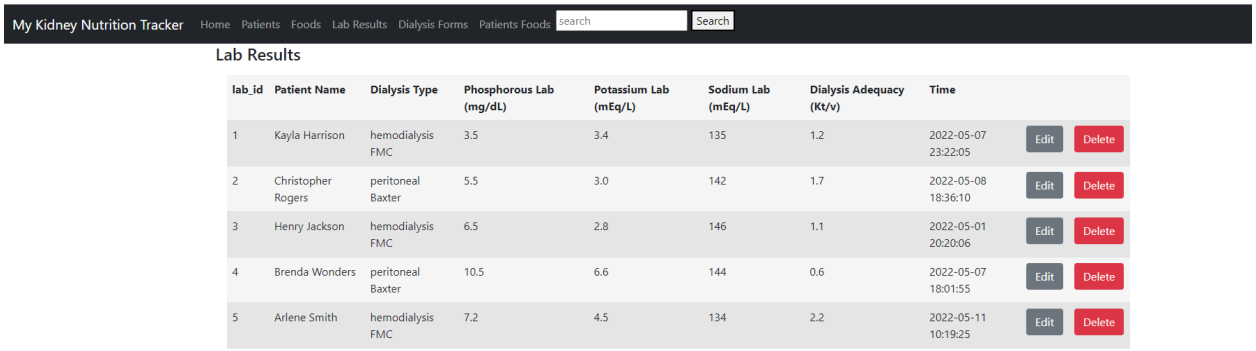
Yes No

SEARCH Patients page



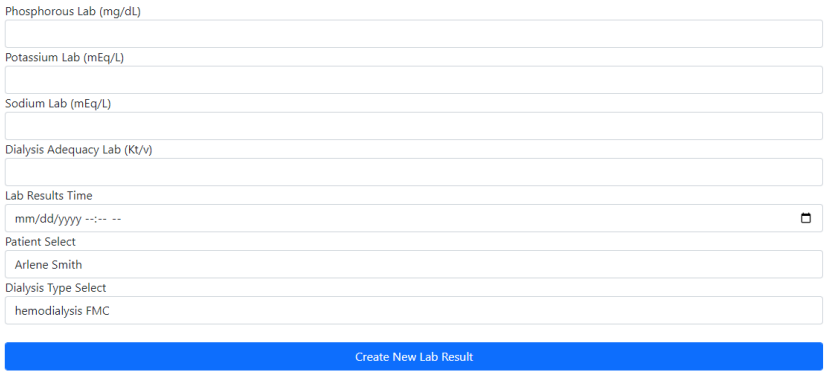
Lab Results Page:

READ/BROWSE/DISPLAY Lab Results page



CREATE/INSERT/ADD Lab Results page

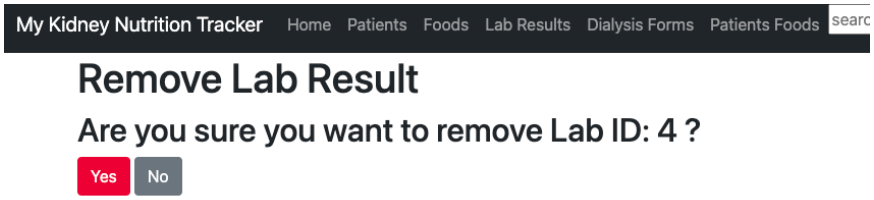
Add New Lab Results



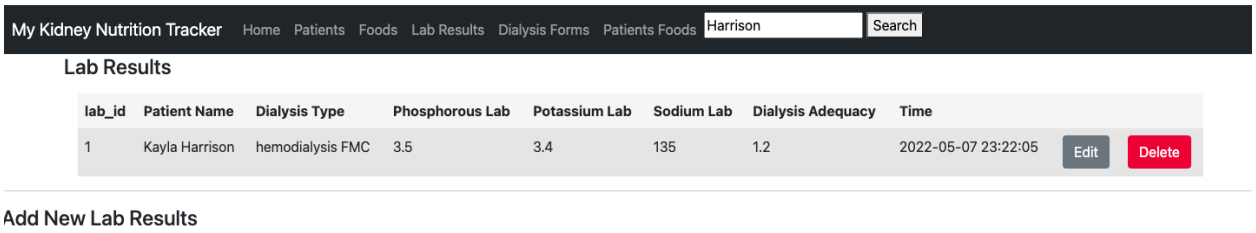
UPDATE Lab Results page

Edit Lab Results

DELETE Lab Results page

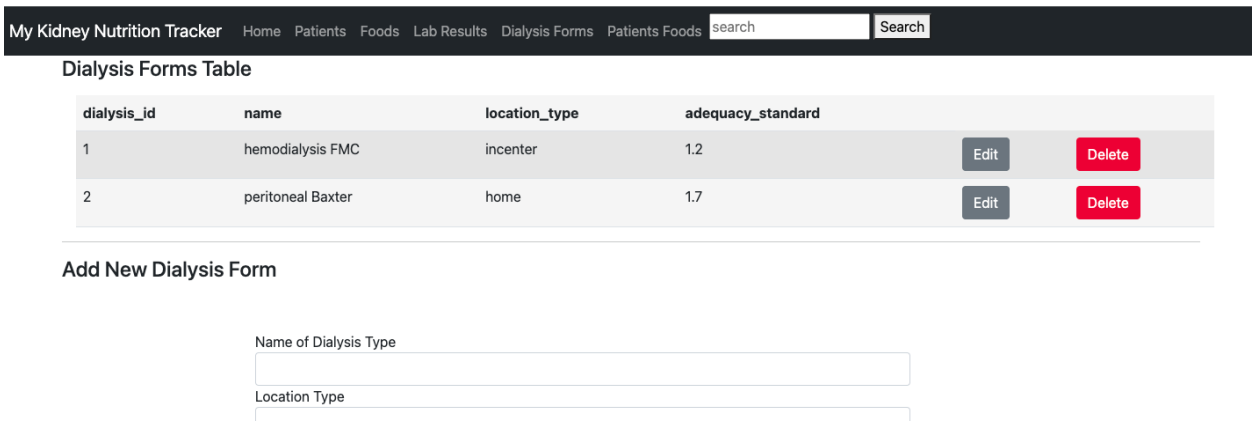


SEARCH Lab Results page

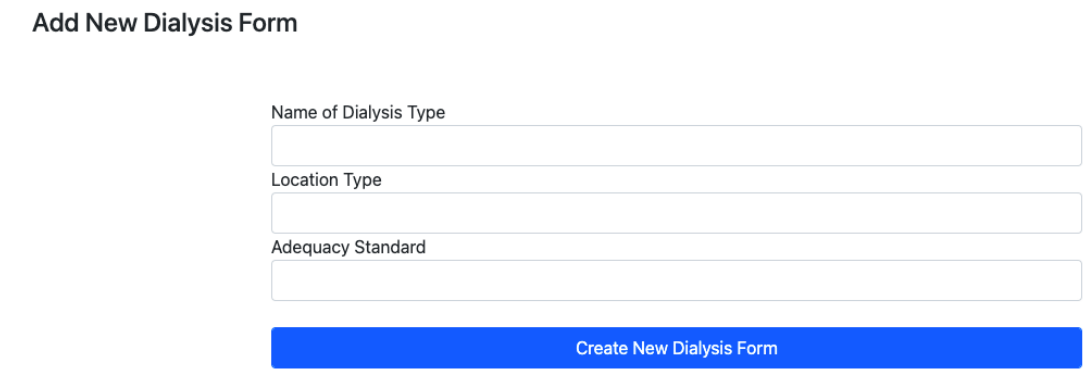


Dialysis Forms Page:

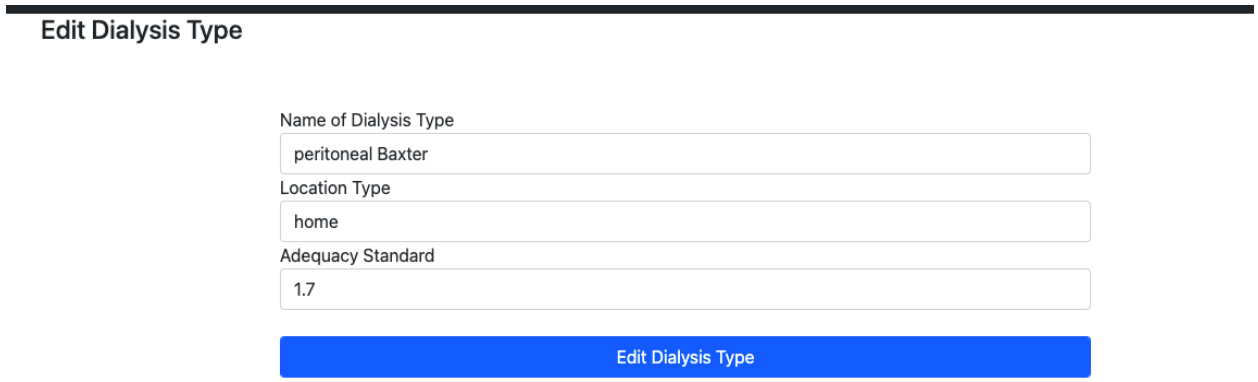
READ/BROWSE/DISPLAY Dialysis Forms page



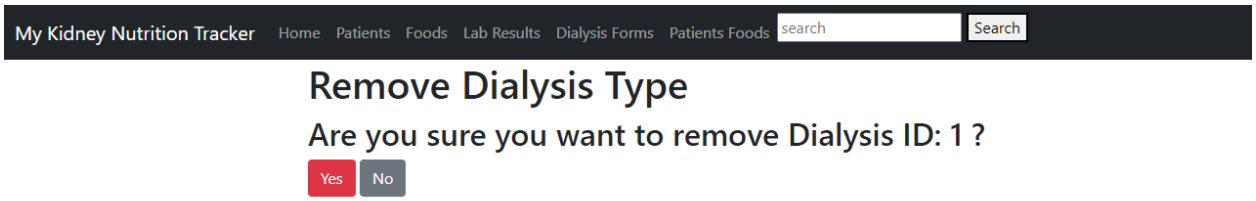
CREATE/INSERT/ADD New Dialysis Forms page



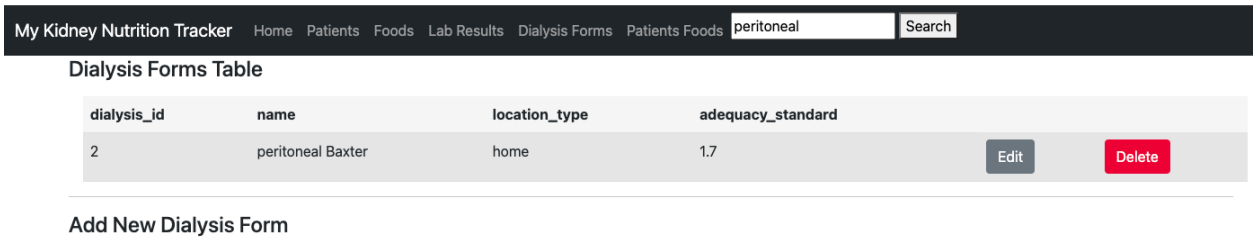
UPDATE New Dialysis Forms page



DELETE Dialysis_Forms page

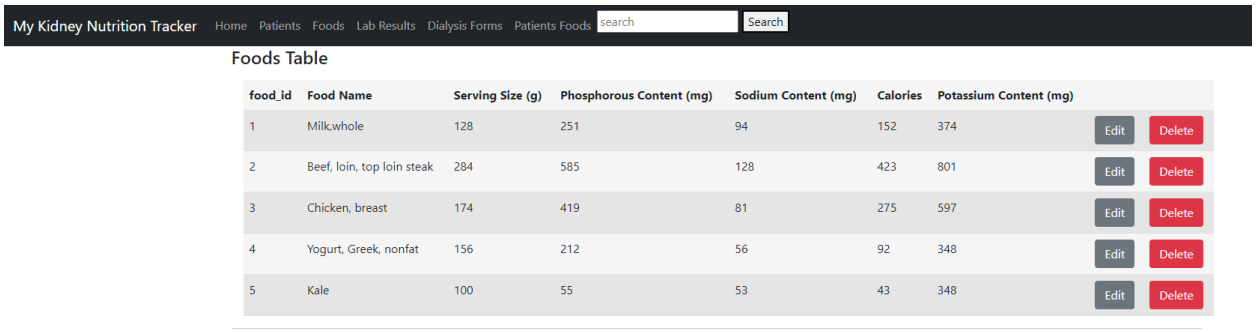


SEARCH Dialysis Forms page

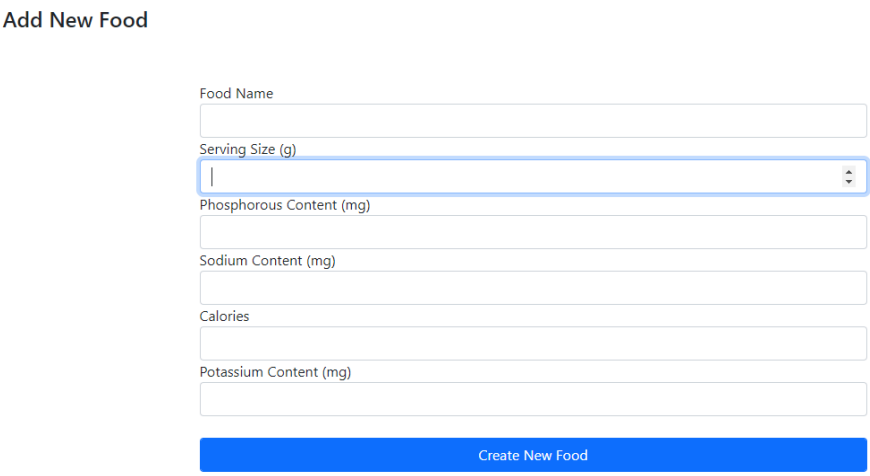


Foods Page:

READ/BROWSE/DISPLAY Foods page



CREATE/INSERT/ADD New Foods page



UPDATE Foods page

Edit Food

Food Name

Beef, loin, top loin steak

Serving Size (g)

284

Phosphorous Content (mg)

585

Sodium Content (mg)

128

Calories

423

Potassium Content (mg)

801

Update Food

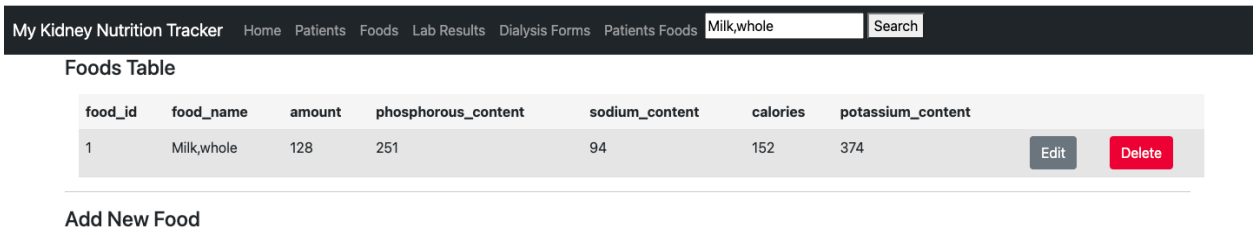
DELETE Foods page

Remove Food

Are you sure you want to remove Food ID: 5 ?

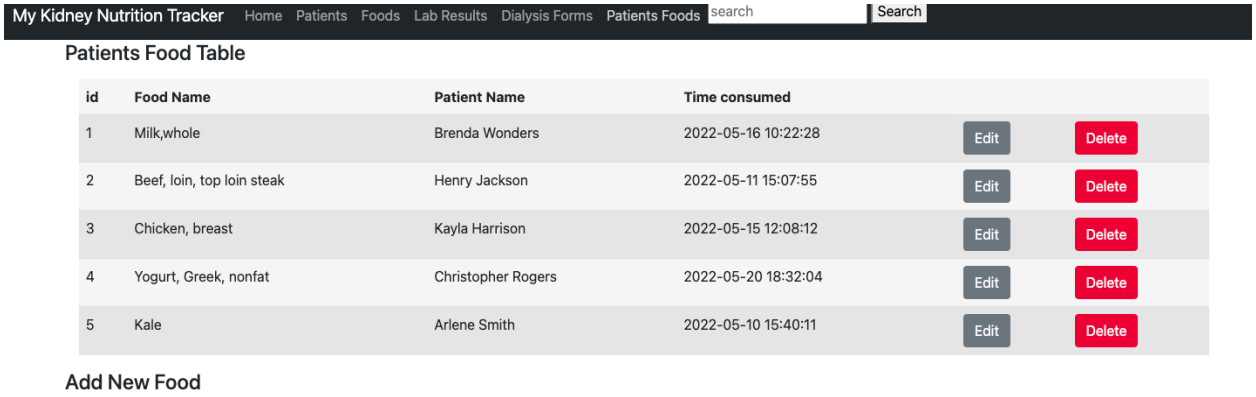
Yes No

Search Foods page

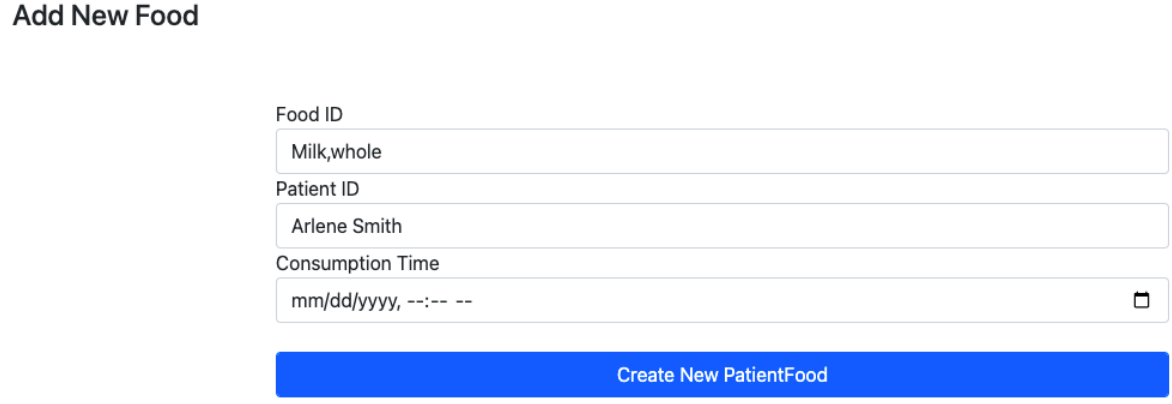


Patients Food Page:

READ/BROWSE/DISPLAY Patients Food page

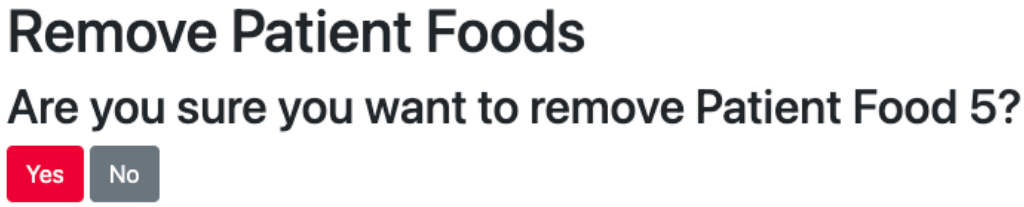


CREATE/INSERT/ADD New Patients Food page



UPDATE New Patients Food page

DELETE New Patients Food page



Citations of Source Code:

Database:

Group 70_Data_Def_Queries file

Exploration-Intro to SQL; Backing up Restoring DB

Date: 5/10/2022

Adapted from /OR/ Based on:

Source URL:
https://canvas.oregonstate.edu/courses/1870053/pages/exploration-intro-to-sql?module_item_id=22036001

https://canvas.oregonstate.edu/courses/1870053/pages/activity-2-backing-up-and-restoring-your-database?module_item_id=22036006

Group 70_Data_Manipulation_Queries

1. #Flask Starter App Guide: Step 6 and 7
2. # Date: 5/20/2022
3. # Adapted from /OR/ Based on:
4. # Source URL:
https://github.com/osu-cs340-ecampus/flask-starter-app#step-6---adding-queries-to-your-app-and-displaying-data

https://github.com/osu-cs340-ecampus/flask-starter-app#step-7---building-a-basic-crud-app

Templates and Pages:

Base.html

1. Python Hosted Flask Bootstrap
2. # Date: 5/12/2022
3. # Adapted from /OR/ Based on:
4. # Source URL https://pythonhosted.org/Flask-Bootstrap/basic-usage.html

delete_dialysis_forms.html
delete_foods.html
delete_labs.html
delete_patient_foods.html
delete_patient.html

1. #Plural Sight and Flask Starter App
2. # Date: 5/2/2022
3. # Adapted from /OR/ Based on:
4. # Source URL
https://app.pluralsight.com/library/courses/structuring-growing-flask-application/

https://github.com/osu-cs340-ecampus/flask-starter-app#delete

foods.html
index.html
lab_results.html
patients_foods.html
patients.html
dialysis_forms.html

1. # Plural Sight and Flask Starter App
2. # Date: 5/21/2022
3. # Adapted from /OR/ Based on:
4. # Source URL
https://app.pluralsight.com/library/courses/structuring-growing-flask-application/
https://github.com/osu-cs340-ecampus/flask-starter-app#step-4---templates

Also based code on:

Grinberg, M. (2018). Flask web development: developing web applications with python. " O'Reilly Media, Inc.", Chapter 3

search.html

1. # w3 schools
2. # Date: 6/1/2022
3. # Adapted from /OR/ Based on:
4. # Source URL https://www.w3schools.com/howto/howto_css_searchbar.asp

update_dialysis_type.html
update_food.html
update_lab_results.html
update_patient.html
update_patients_food.html

1. # Flask Starter App
2. # Date: 5/21/2022
3. # Adapted from /OR/ Based on:
4. # Source URL
https://github.com/osu-cs340-ecampus/flask-starter-app#updateedit

tracker.py

1. # Flask Starter App, Wtforms, Grinberg Flask Web Dev Book
2. # Date: 5/1/2022
3. # Adapted from /OR/ Based on:

4. # Source URL

Grinberg, M. (2018). Flask web development: developing web applications with python.
https://github.com/osu-cs340-ecampus/flask-starter-app
https://wtforms.readthedocs.io/en/3.0.x/