VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CBS1005 Software Engineering Methodologies

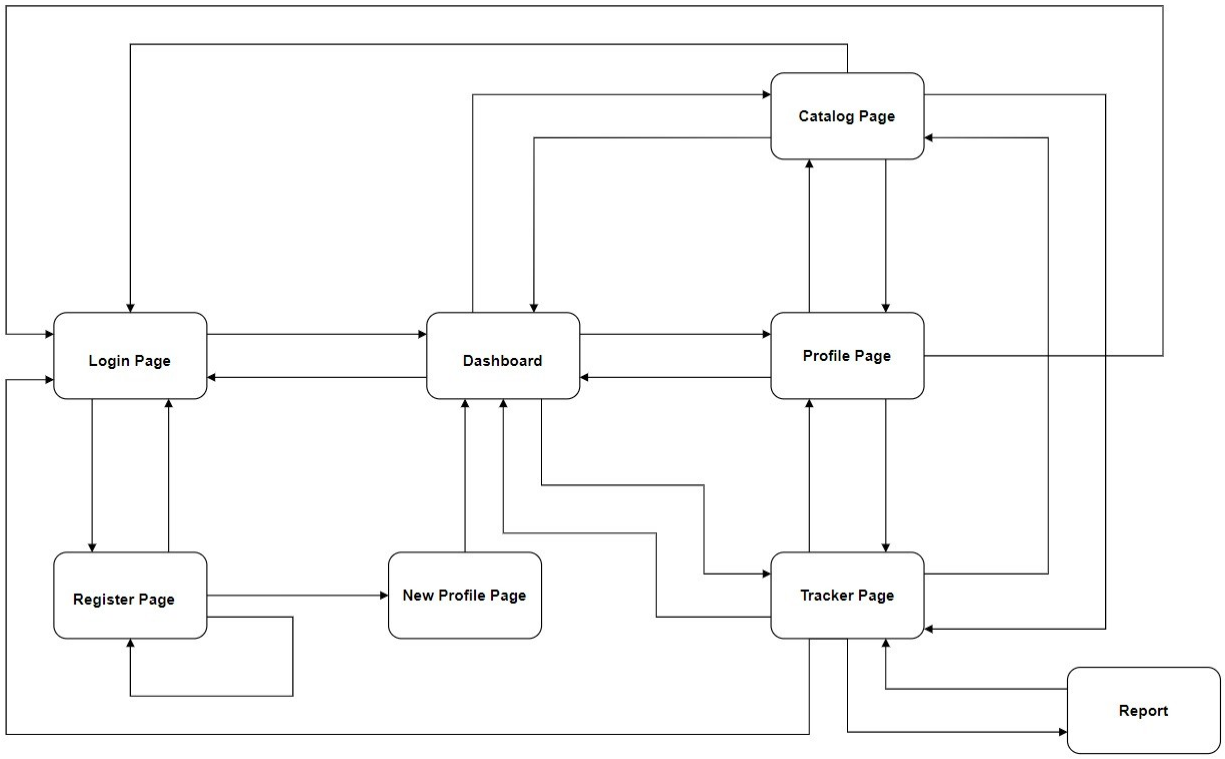
Lab Assessment – Implementation & Testing

Name: Omkar Kabadagi
Registration Number:19BBS0038
Slot: L59+L60
Date:04/10/2020

Team Members:
Rithvik Ayithapu Sai – 19BBS0031
Omkar Kabadagi – 19BBS0038
Akhil Chaitanya Ghanta – 19BBS0039

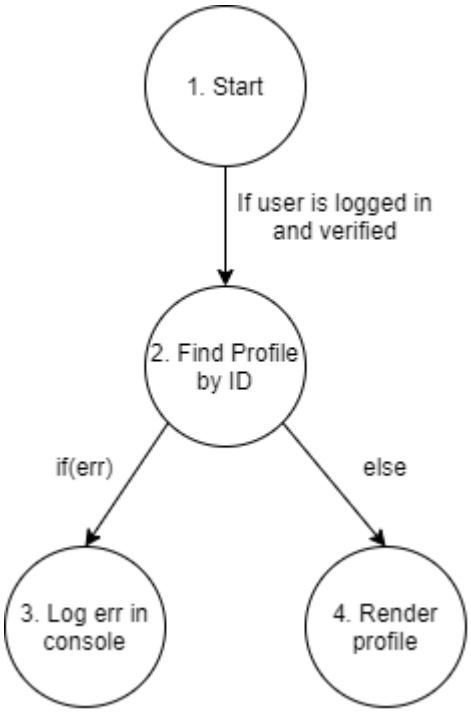
1. Introduction

1.1 Block Diagram



1.2 Routes/Modules

1. **/profile –**
    1. Start and Show profile – Pseudocode:
    if (user is logged in and verified) {
        fetch profile using id and render it;
    } else {
        Redirect to "/profile/new";
    }

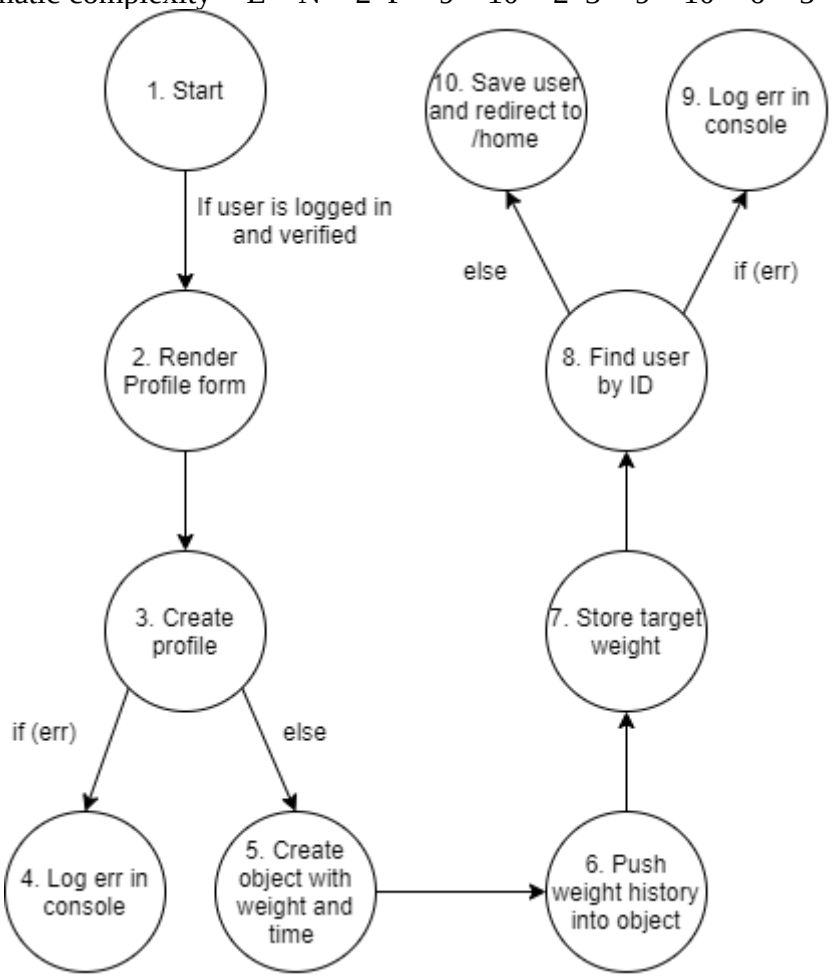


Cyclomatic complexity = E – N + 2*P
Where,
E = number of edges in the flow graph,
N = number of nodes in the flow graph,
P = number of nodes that have exit points
So cyclomatic complexity = 3 – 4 – 2*2 = 3 – 4 – 4 = 3

2. Create new profile – Pseudocode:
Render form for user input;
Create profile from user input;
If (error is generated) {
    Log the error in console;
} else {
    Create object with weight of user and timestamp;
    Push weight history of user;
    Record target weight of user;
    Save profile;
    Fetch profile of user {
        If (error is generated) {

Log the error in console;
} else {
Redirect to “/home”;
}
}

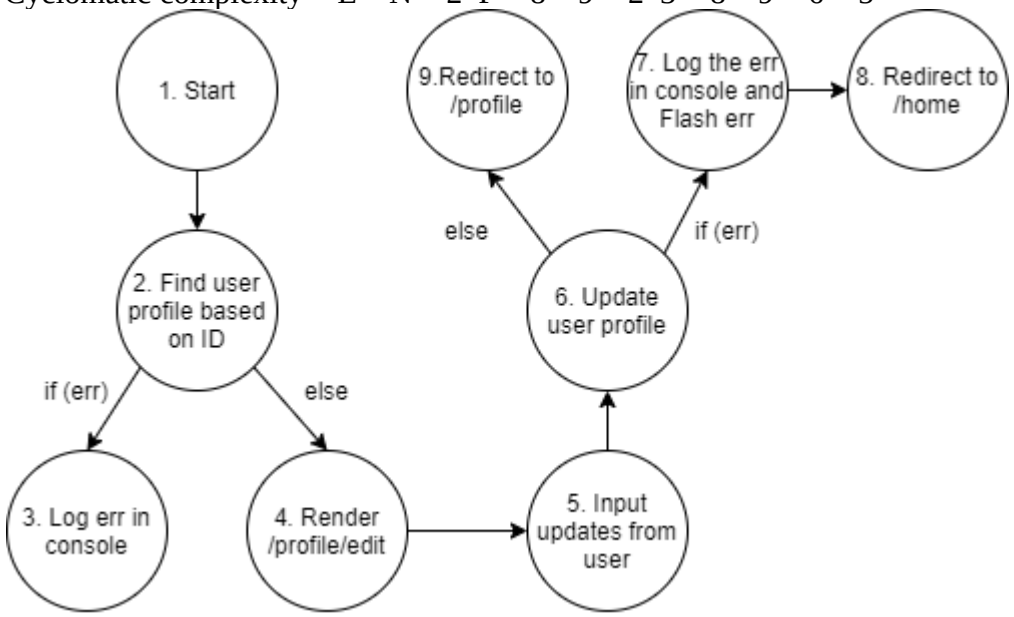
Cyclomatic complexity = E – N + 2*P = 9 – 10 + 2*3 = 9 – 10 + 6 = 5



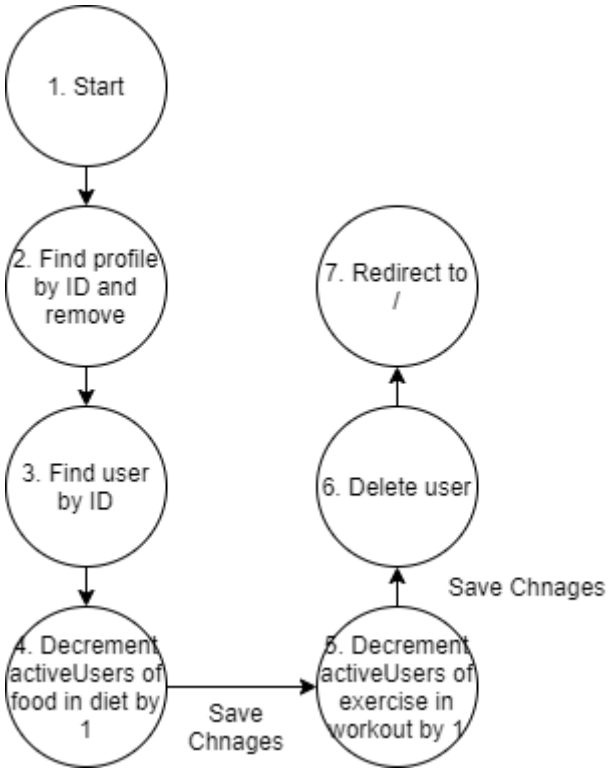
3. Edit profile – Pseudocode:

```
Fetch profile of user;
If (error is generated) {
    Log the error in console;
} else {
    Render profile of user;
}
Input changes user wants to make in profile;
Update profile;
If (error is generated) {
    Log the error in console;
Redirect “/home”;
} else {
```

Redirect “/profile”;
}
Cyclomatic complexity = E – N + 2*P = 8 – 9 + 2*3 = 8 – 9 + 6 = 5



4. Delete profile – Pseudocode:
Fetch user profile;
Find user’s diet;
Reduce active users in diets by 1;
Reduce active users in workouts by 1;
Remove user profile;
Save changes;
If (error is generated) {
Log the error in console;
} else {
Redirect to “/home”;
}
Cyclomatic complexity = E – N + 2*P = 6 – 7 + 2*1 = 1

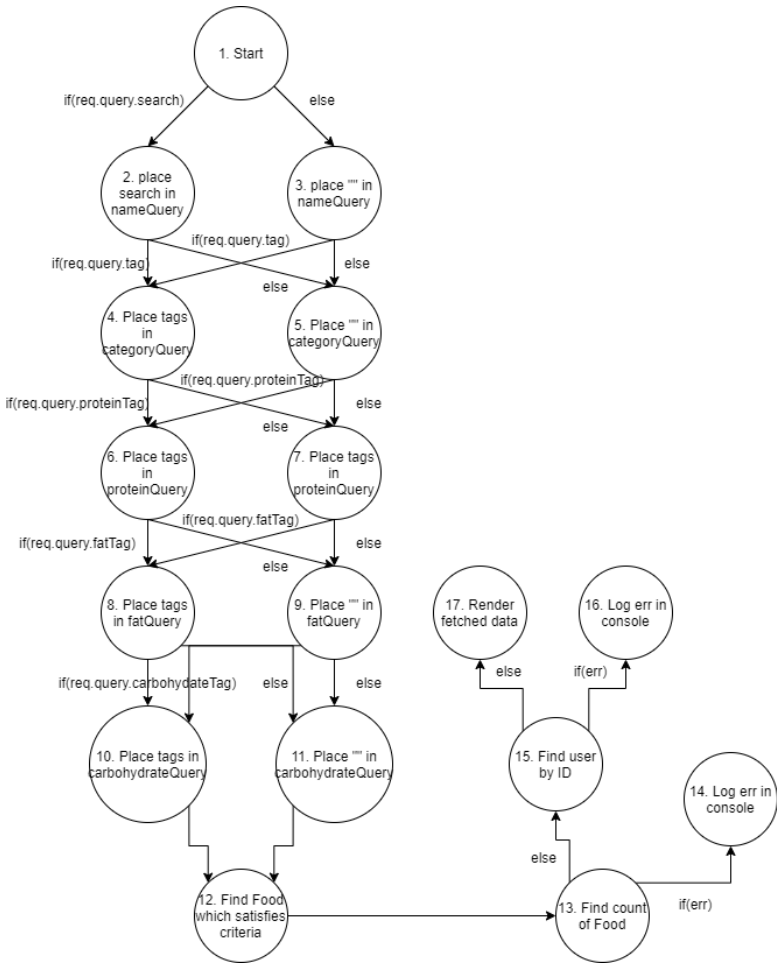


2. /diets –

1. Start
2. Show dishes – Pseudocode:

```
Fetch dishes and render them;
If (search is not used) {
    Render all dishes;
} else {
    Search for given keyword;
    Render the dishes found;
}
If (category is selected) {
    Fetch dishes of specific category;
    Render dishes;
}
If (nutrient is selected) {
    Fetch dishes having specific category;
    Render dishes;
}
If (error is generated) {
    Log the error in console;
}
```

Cyclomatic complexity = E – N + 2*P = 25 – 17 + 2*3 = 8 + 6 = 14

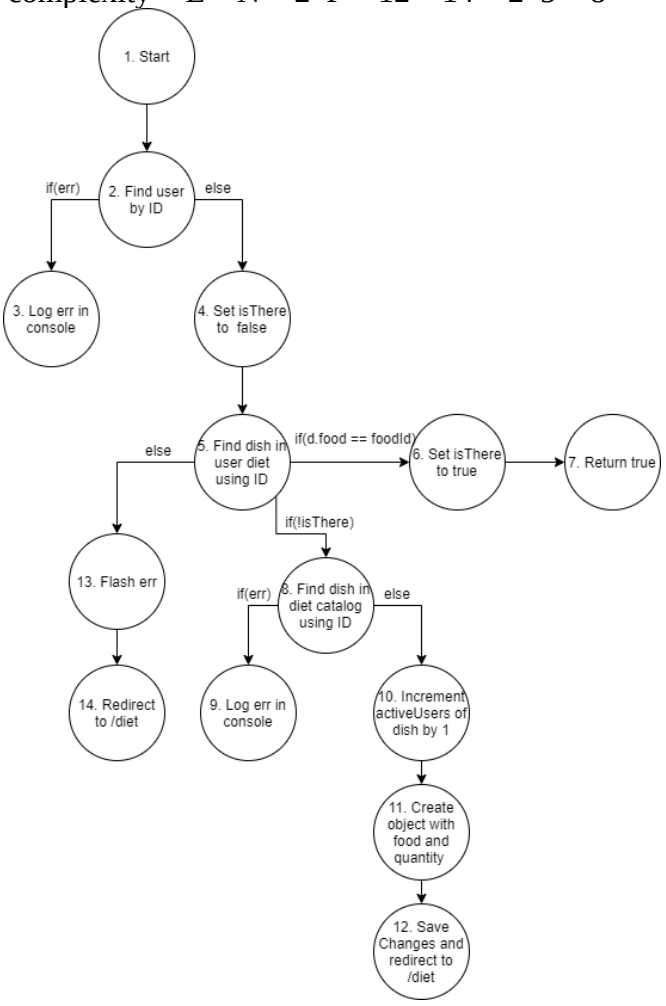


3. Add diet – Pseudocode:

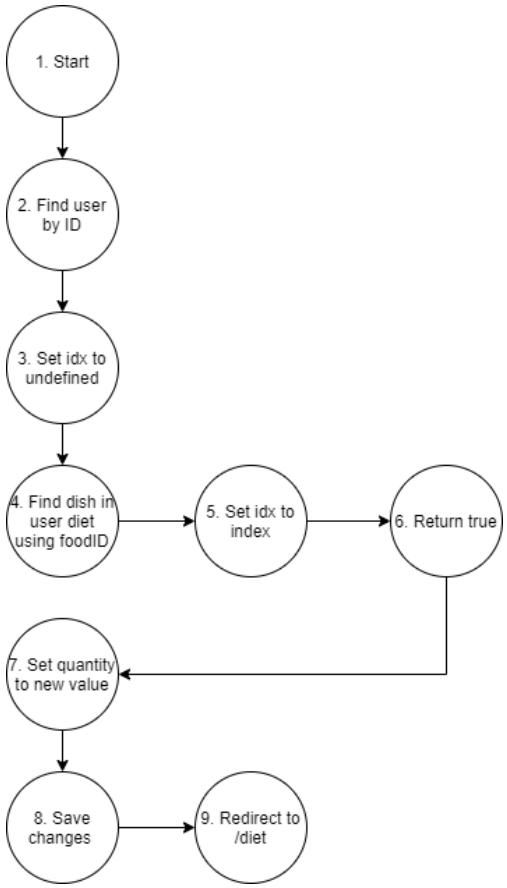
```
Fetch user details;
If (error is generated) {
    Log the error in console;
} else {
    isThere = false;
    Check diet of user;
    If (user has dish already in diet) {
            Set isThere = true;
    }
    If (!isThere) {
            Fetch the required dish;
            If (error is generated) {
                    Log the error in console;
            } else {
                    Increment activeUsers of dish by 1;
                    Save changes;
                    Create object storing food and quantity;
                    Push the diet to user;
```

```
                    Save changes;
                    Redirect to "/diet";
                }
        } else {
                Render error = "Already present in checkout list";
                Redirect to "/diet";
        }
}
```

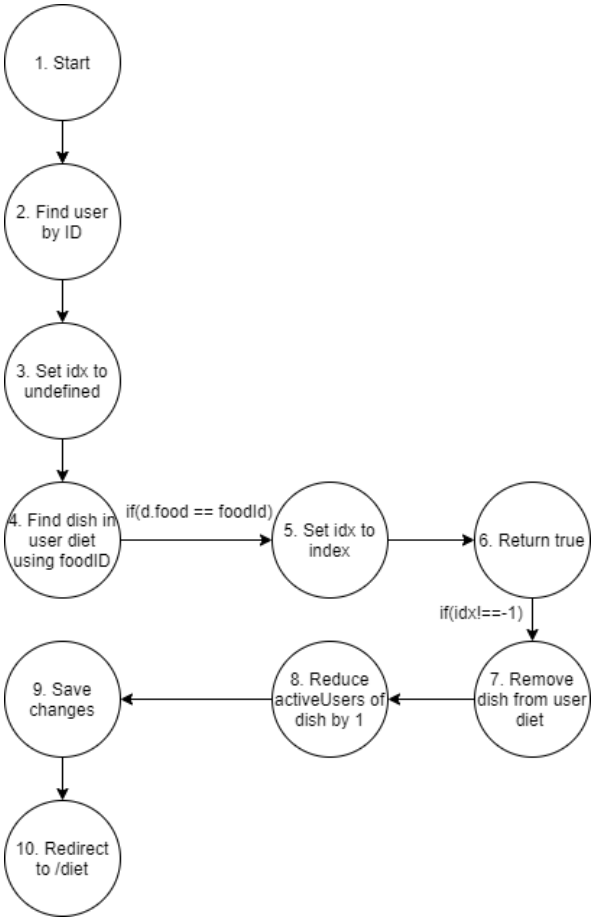
Cyclomatic complexity = E – N + 2*P = 12 – 14 + 2*5 = 8



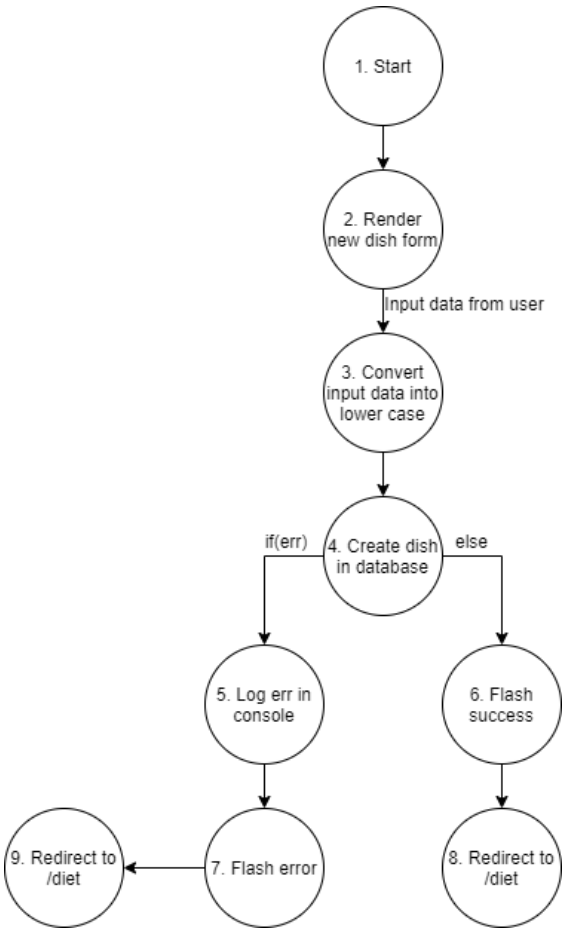
4. Change diet – Pseudocode:
   Fetch user and dish details;
   Fetch the diet of the user;
   Change quantity of the dish in diet;
   Save changes;
   Redirect to "/diet";
   Cyclomatic complexity = E – N + 2*P = 8 – 9 + 2*1 = 8 – 9 + 2 = 1



5. Remove diet – Pseudocode:
   Fetch user and dish details;
   Fetch the diet of the user;
   Remove dish from the list of dishes in diet;
   Save changes;
   Reduce activeUsers of dish by 1;
   Save Changes;
   Redirect to “/diet”;
   Cyclomatic complexity = E – N + 2*P = 9 – 10 + 2*1 = 9 – 10 + 2 = 1

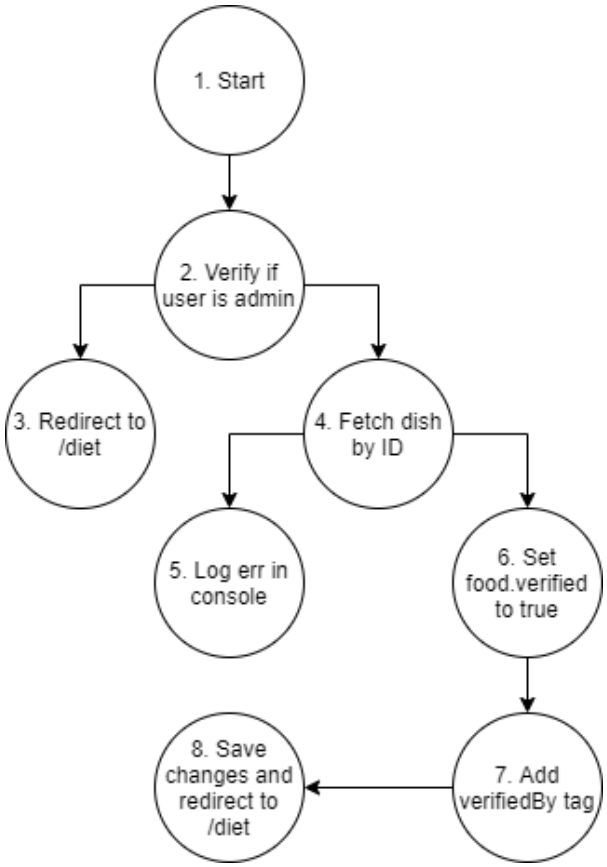


6. Add new dish – Pseudocode:
   Render the form for taking input of the new dish;
   Create new dish;
   If (error is generated) {
       Render “Something went wrong”;
       Redirect to “/diet”;
   } else {
       Render “Successfully submitted”;
       Redirect to “/diet”;
   }
       Cyclomatic complexity = E – N + 2*P = 8 – 9 + 2*2 = 8 – 9 + 4 = 3



7. Verify dish – Pseudocode:

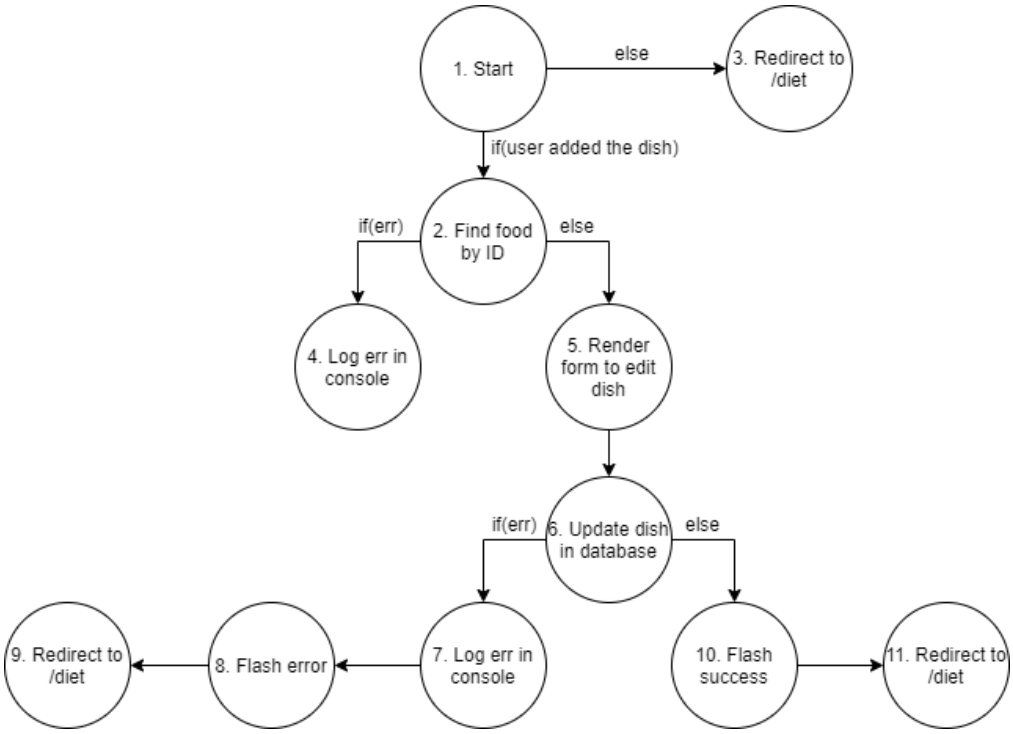
```
If (user is admin) {
    Fetch food using id;
    If (error is generated) {
            Log the error in console;
    } else {
            Set verified = true;
            Set verifiedBy = user;
            Save changes;
            Redirect to "/diet";
    }
    Cyclomatic complexity = E – N + 2*P = 7 – 8 + 2*3 = 7 – 8 + 6 = 5
```



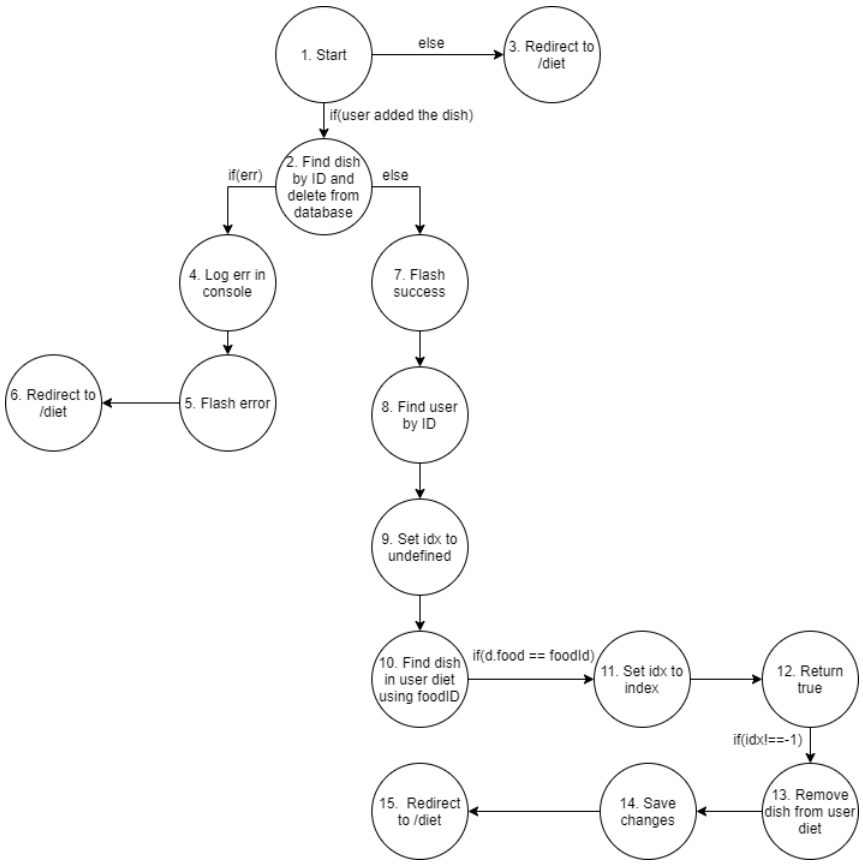
8. Edit and update dish – Pseudocode:

```
Fetch food by id;
If (error is generated) {
    Log the error in console;
} else {
    Render “diet/edit”;
}
Input the update to the dish;
Update the dish;
If (error is generated) {
    Log the error in console;
    Flash “something went wrong”;
    Redirect to “/diet”;
} else {
    Flash “Update successful”;
    Redirect “/diet”;
}
```

Cyclomatic complexity = E – N + 2*P = 10 – 11 + 2*4 = 10 – 11 + 8 = 7



9. Delete dish – Pseudocode:
   Find dish by its id and delete it;
   If (error is generated) {
       Log the error in console;
       Flash “something went wrong”;
       Redirect to “/diet”;
   } else {
       Flash “Food deleted”;
       For each user {
           Fetch diets;
           Remove the dish from diet;
           Save changes;
       }
       Redirect to “/diet”;
   }
       Cyclomatic complexity = E – N + 2*P = 14 – 15 + 2*3 = 14 – 15 + 6 = 5

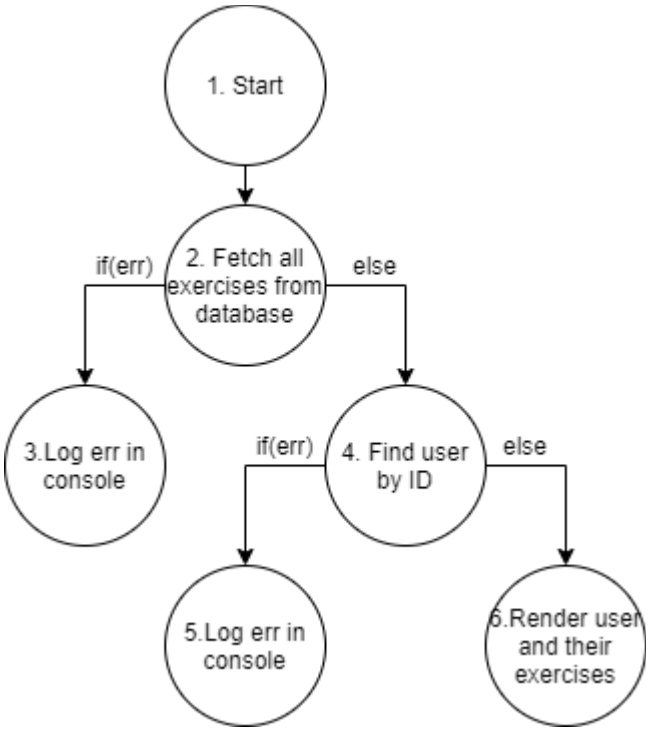


3. /workout

1. Start
2. Show Workouts – Pseudocode:

```
Fetch all workouts and render on screen;
 If (error is generated) {
     Log the error in console;
}
Fetch workouts of the user and render on screen;
If (error is generated) {
     Log the error in console;
}
```

Cyclomatic complexity = E – N + 2*P = 5 – 6 + 2*3 = 5 – 6 + 6 = 5



3. Add Workouts – Pseudocode:

```
Fetch user profile using ID;
If (error is generated) {
    Log the error in console “User not found”;
} else {
    Set isThere to false;
    Find exercise in the user’s profile using id;
    If (exercise found) {
        Set isThere to true;
        Return true;
    }
    If (isThere is false) {
        Find exercise by ID in exercises list;
        If (error is generated) {
            Log the error in console “Exercise not found”;
        }
        Else {
            Increment activeUsers of exercise by 1;
            Save the changes;
            Create object {
                Exercise: exercise,
                Duration: input duration
            }
            Push the object to user profile;
            Save the changes;
            Redirect to “/workout”;
```

}

Else {

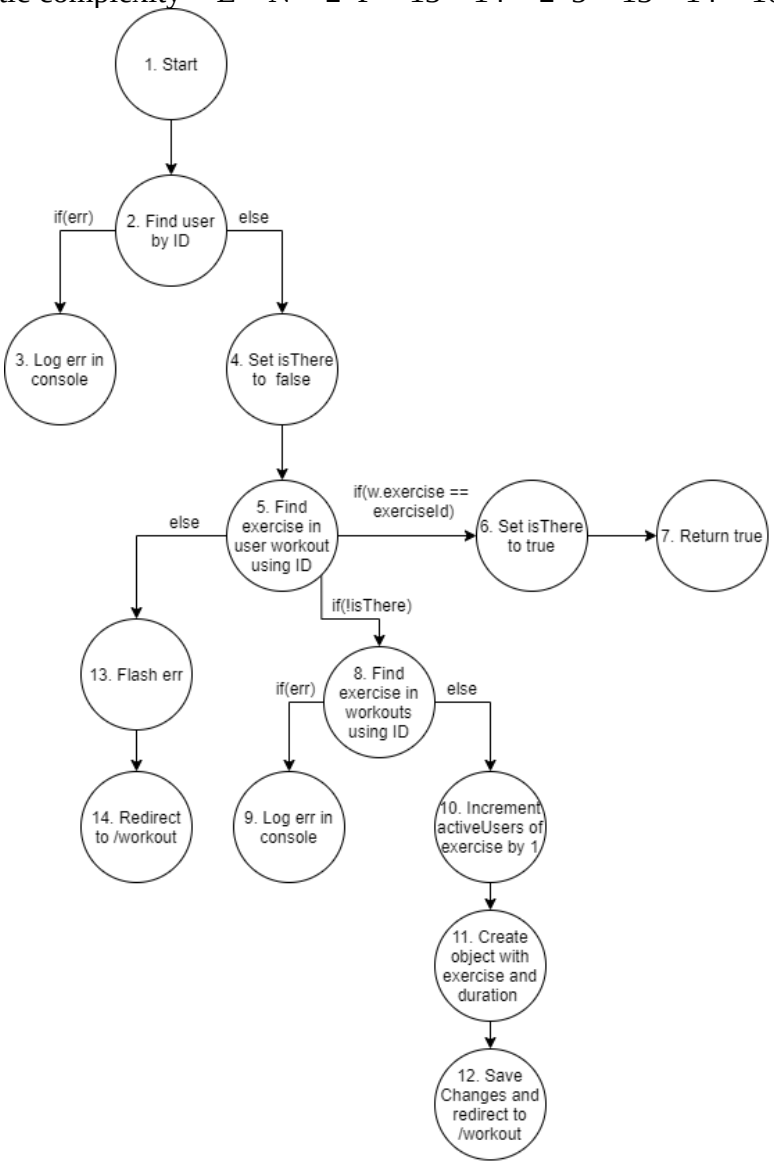
Flash error “Already present in checkout list”;

Redirect to “/workout”;

}

}

Cyclomatic complexity = E – N + 2*P = 13 – 14 + 2*5 = 13 – 14 + 10= 9



4. Remove Workouts – Pseudocode:

Fetch exercises of the user using ID;

Set idx = undefined;

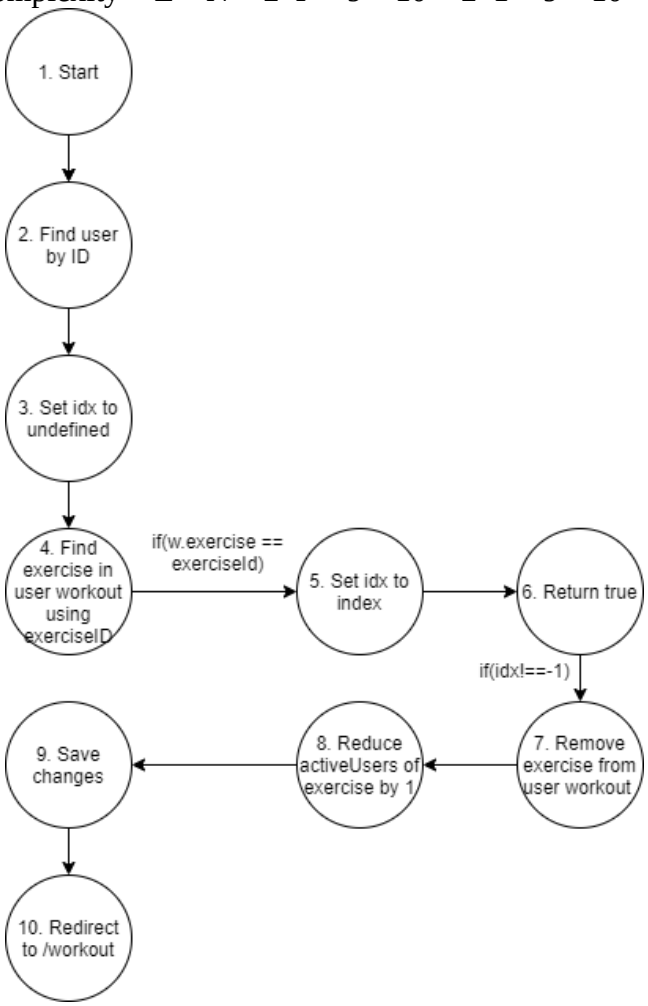
If (exercise index found in user profile) {

Set idx = index;

Return true;

}

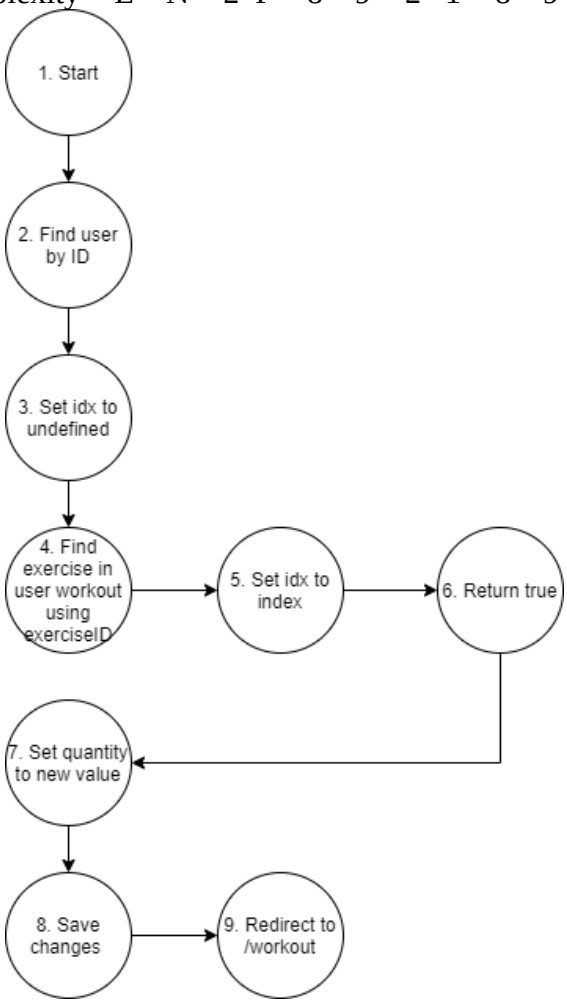
If (idx is not equal to -1) {
    Remove exercise from array of user workout;
    Save changes;
    Find exercise by ID in list of exercises;
    Increment activeUsers of exercise by 1;
    Save changes;
    Redirect to “/workout”;
}
    Cyclomatic complexity = E – N + 2*P = 9 – 10 + 2*1 = 9 – 10 + 2 = 1



5. Change Workouts – Pseudocode:
    Fetch user profile using ID;
    Set idx = undefined;
    Find exercise index in user workout array;
    If (exercise index found) {
        Set idx = index;
        Return true;
    }
    Set duration of workout to input from user;

Save changes;
Redirect to “/workout”;

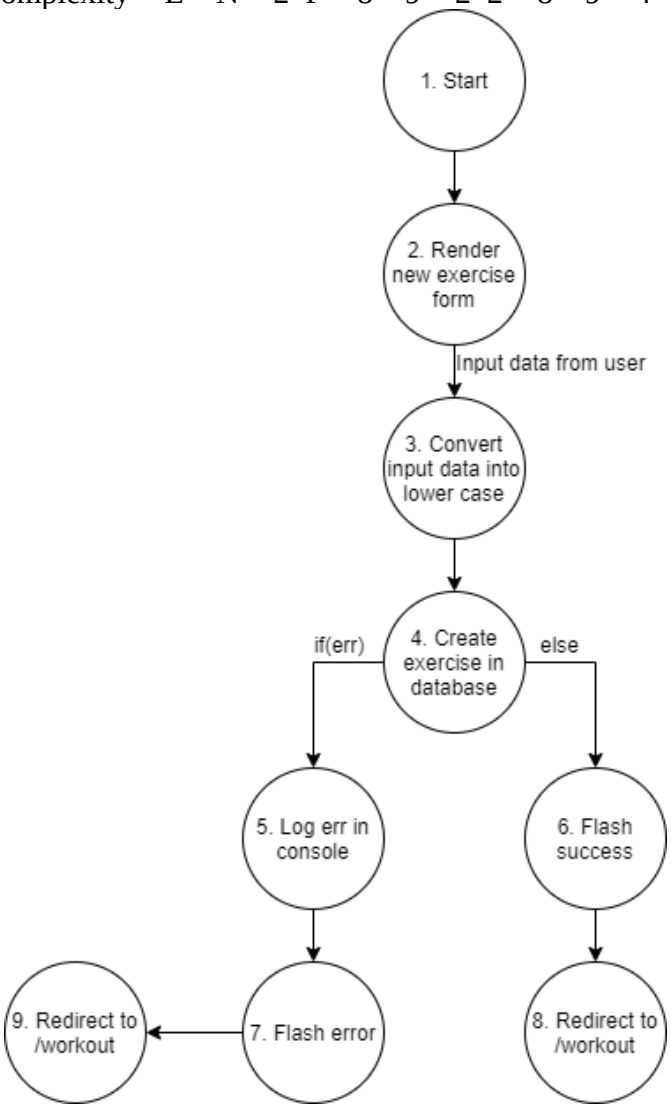
Cyclomatic complexity = E – N + 2*P = 8 – 9 + 2 *1 = 8 – 9 + 2 = 1



6. Create new exercise – Pseudocode:

```
Render form to input of new exercise;
Set name to exercise name input by user;
Set tag to exercise tag input by user;
Set addedBy to user who gave input;
Create exercise {
    If (error is generated) {
            Log the error in console;
            Flash error “Something went wrong”;
            Redirect to “/workout”;
    } else {
            Flash success “Successfully submitted”;
            Redirect to “/workout”;
    }
}
```

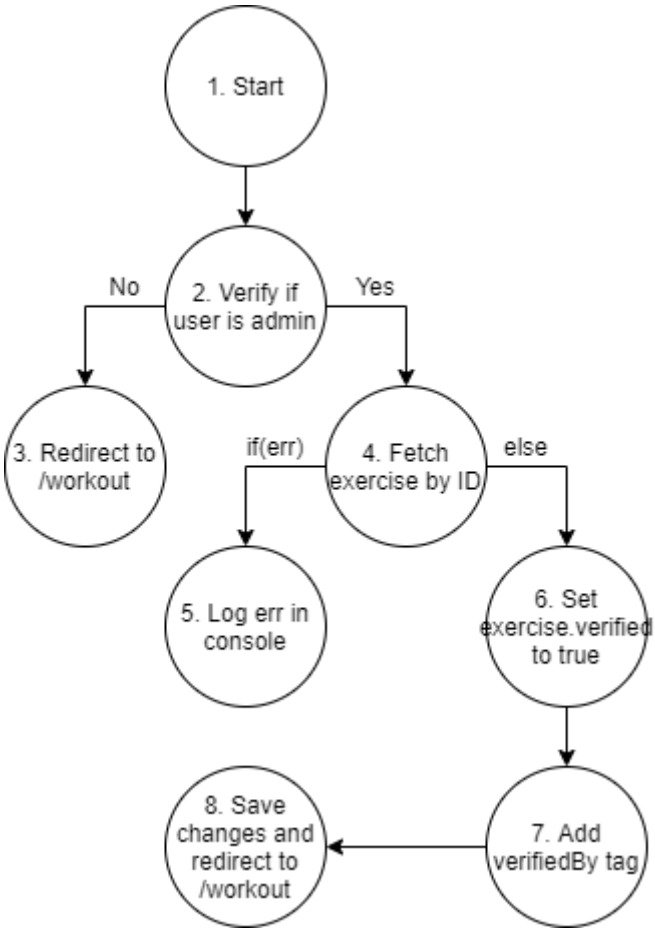
Cyclomatic complexity = E – N + 2*P = 8 – 9 + 2*2 = 8 – 9 + 4 = 3



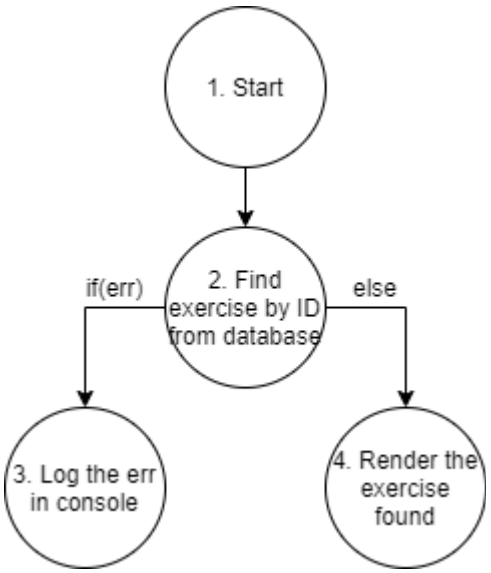
7. Verify exercise – Pseudocode:

```
If (user is admin) {
    Find the exercise by ID;
    If (error is generated) {
            Log the error in console;
    } else {
            Set verified to true;
            Set verifiedBy to user who gave input;
            Save changes;
            Redirect to "/workout";
    }
} else {
    Flash error "User must be admin to verify";
}
```

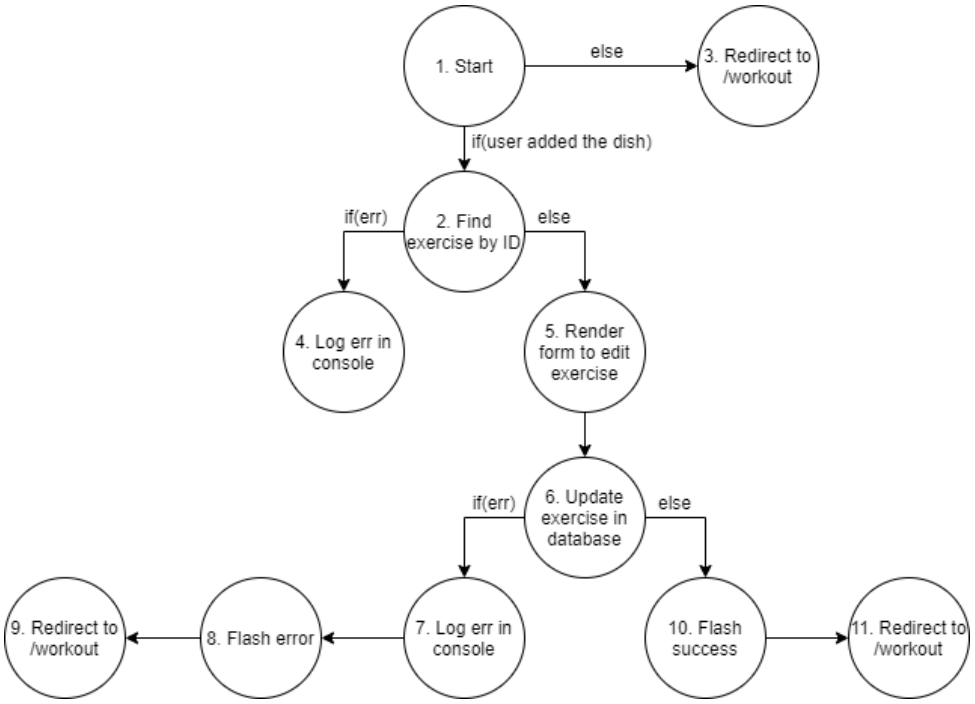
Cyclomatic complexity = E – N + 2*P = 7 – 8 + 2*3 = 7 – 8 + 6 = 5



8. Show exercise – Pseudocode:
   Fetch exercise from exercises list using ID;
   If (error is generated) {
       Log the error in console;
   } else {
       Render exercise that was fetched;
   }
       Cyclomatic complexity = E – N + 2*P = 3 – 4 + 2*2 = 3 – 4 + 4 = 3

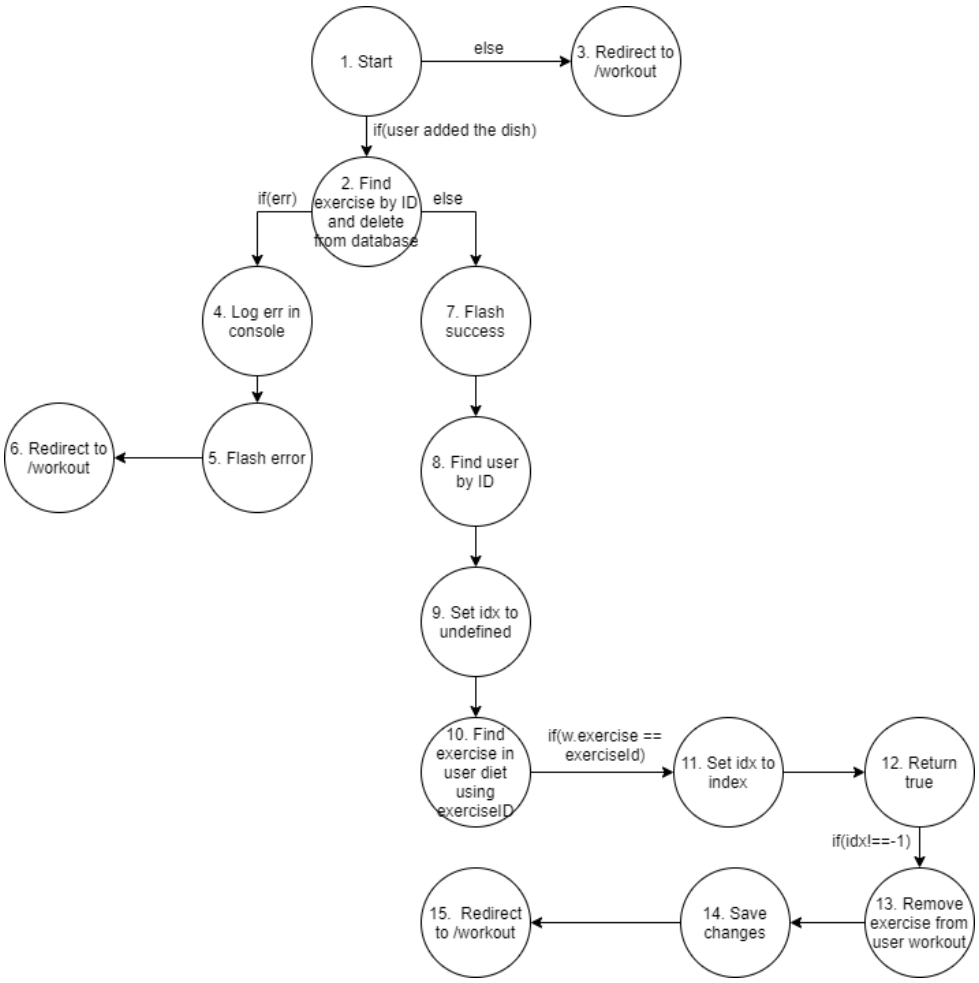


9. Edit and Update exercise – Pseudocode:
   Fetch exercise from exercises list using exid in request;
   Render the fetched exercise;
   Input desired changes from user;
   Update the exercise with given changes;
   If (error is generated) {
       Log the error in console;
       Flash error “Something went wrong”;
       Redirect to “/workout”;
   } else {
       Flash success “Update successful”;
       Redirect to “/workout/exercise/eixd”;
   }
       Cyclomatic complexity = E – N + 2*P = 10 – 11 + 2*4 = 10 – 11 + 8 = 7



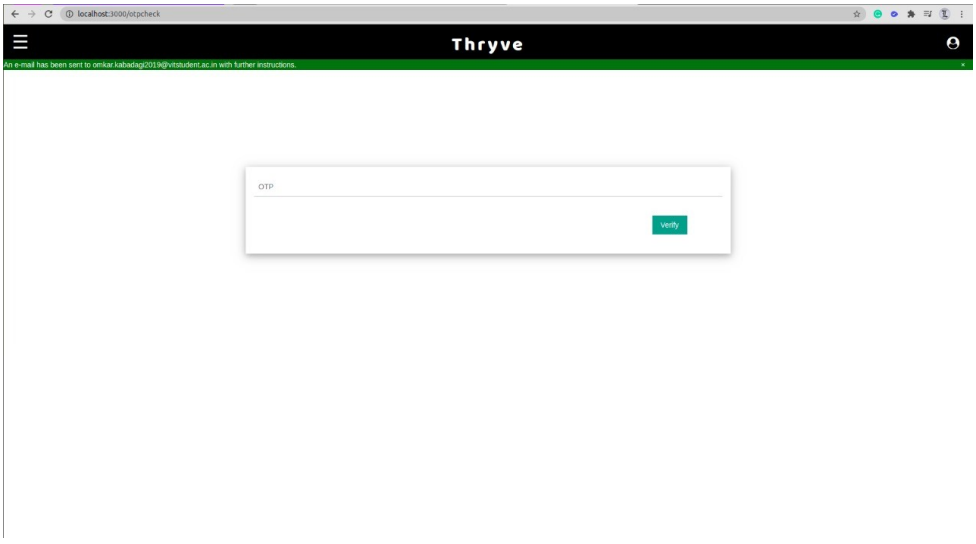
10. Delete exercise – Pseudocode:

```
Use exid in request to find the exercise and delete it;
If (error is generated) {
    Log the error in console;
    Flash error “Something went wrong”;
    Redirect to “/workout”;
} else {
    Flash success “Exercise deleted”;
    For all users {
            Set idx = undefined;
            If (user workout contains deleted exercise) {
                    Index = id of exercise in workout array;
                    Set idx = index;
                    Return true;
            }
    }
    If (idx is not -1) {
            Remove exercise from workouts array;
            Save changes;
    }
    Redirect to “/workout”;
}
    Cyclomatic complexity = E – N + 2*P = 14 – 15 + 2*3 = 14 – 15 + 6 = 5
```

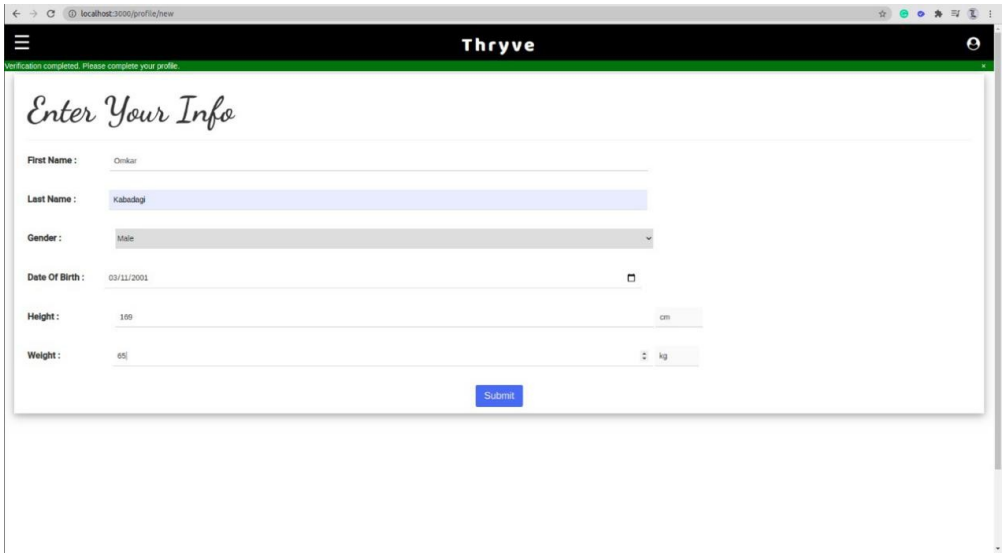


1.3 Screenshots of Modules

1.3.1 Login/Sign Up

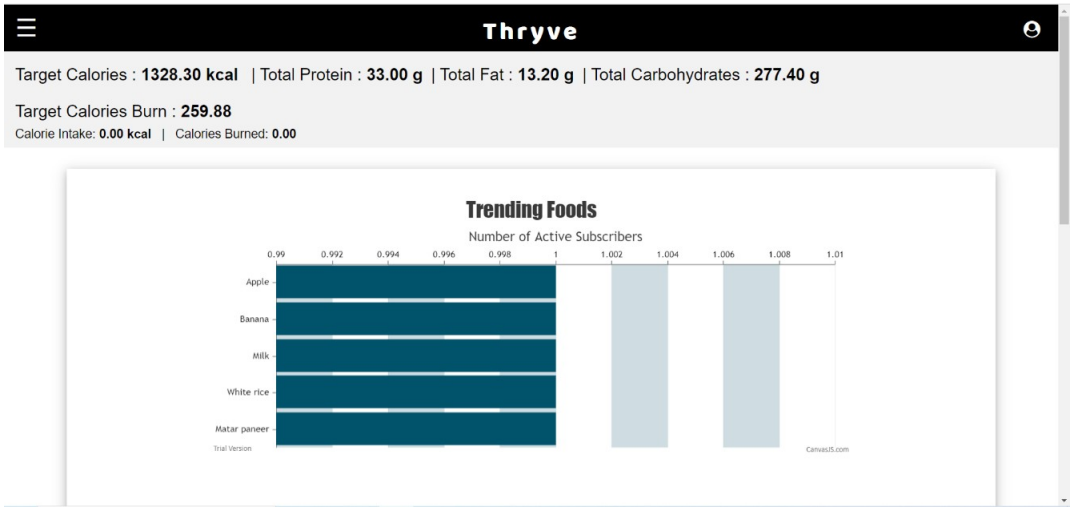


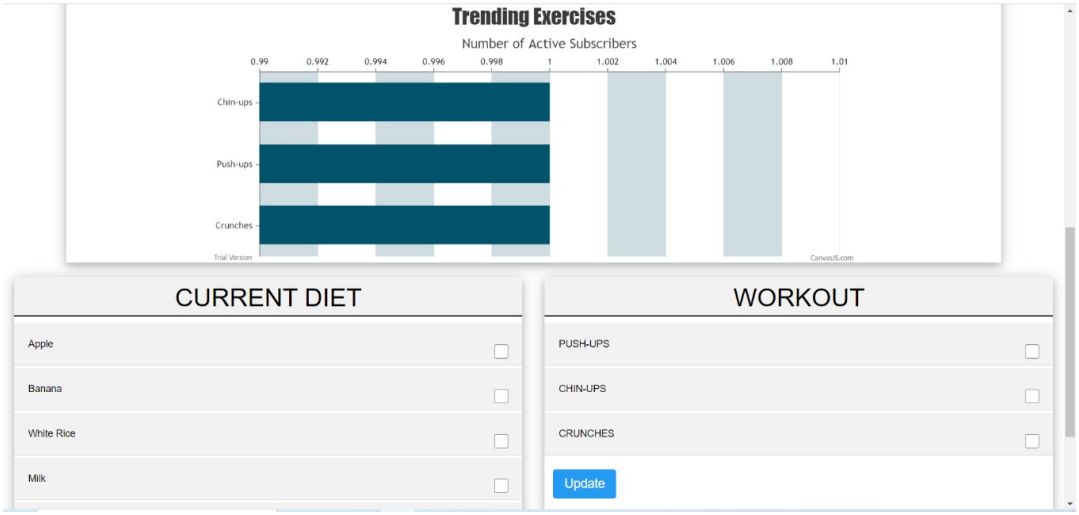
An Email OTP based system is used for those users who either forgot their password or verifying their account for first time.



New users or First time users after successful verification redirected to the profile form page where He/she need to fill their basic details like Name ,Gender ,DOB ,Height and Weight.

1.3.2 Home



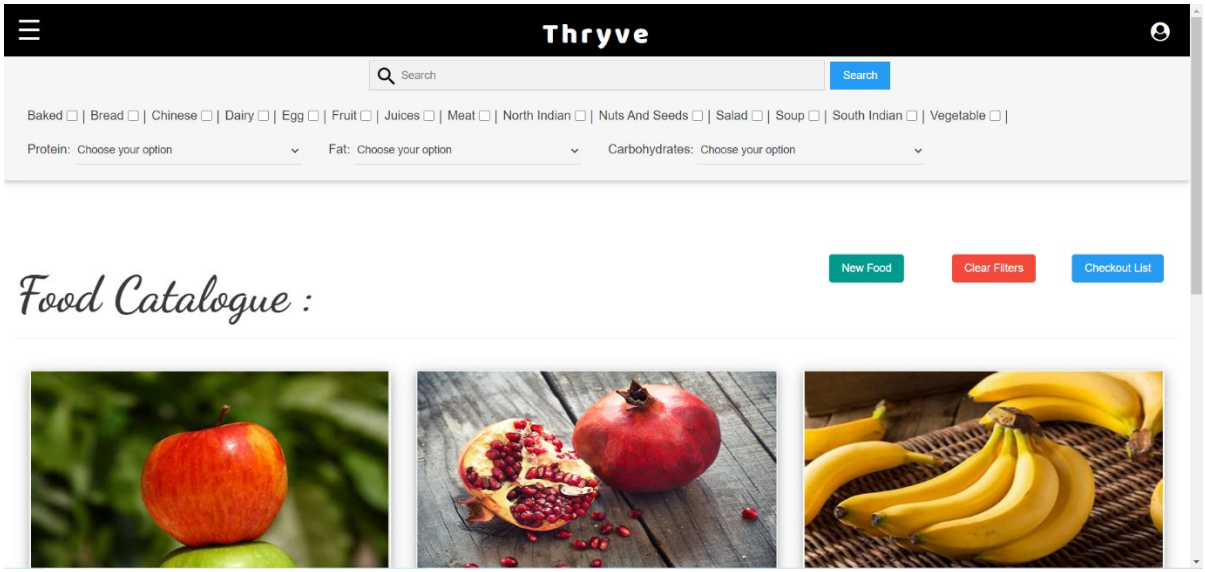


Once the user logs in successfully, he/she redirected to the home page where their nutrient intake is shown along with their calorie goal. Foods and exercises trending among the users is also shown along with the current diet workout of the user. Once a part of the diet is consumed or an exercise is done the user can update it so that the calorie intake or calorie burn at the top of the page is updated.

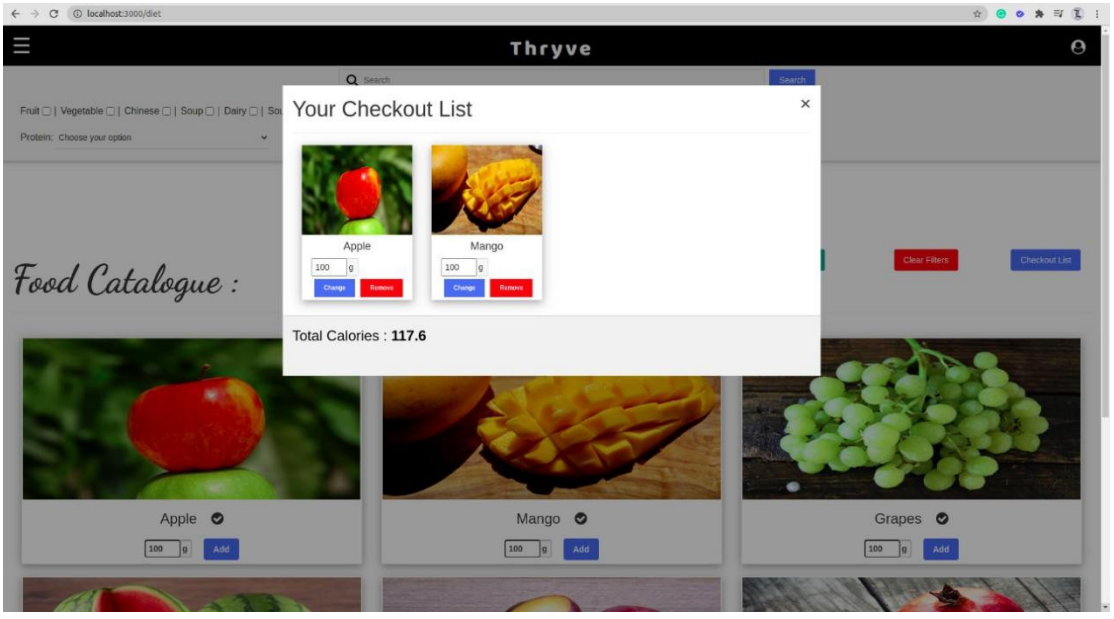
1.3.3 Dashboard

The dashboard is used to jump to the different pages easily.

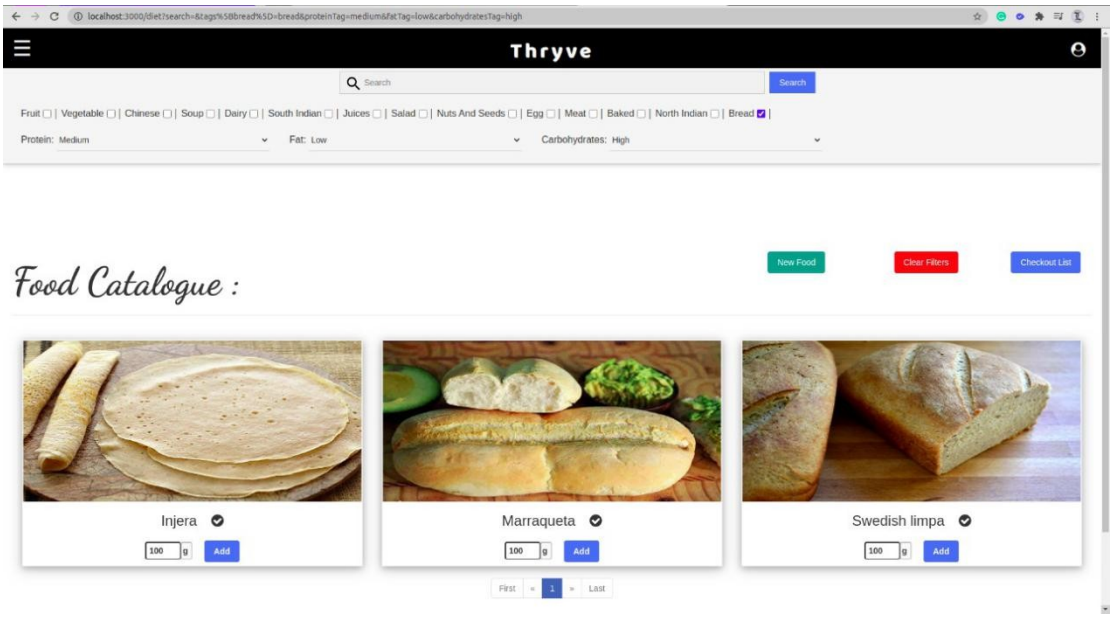
1.3.4 Food Catalogue



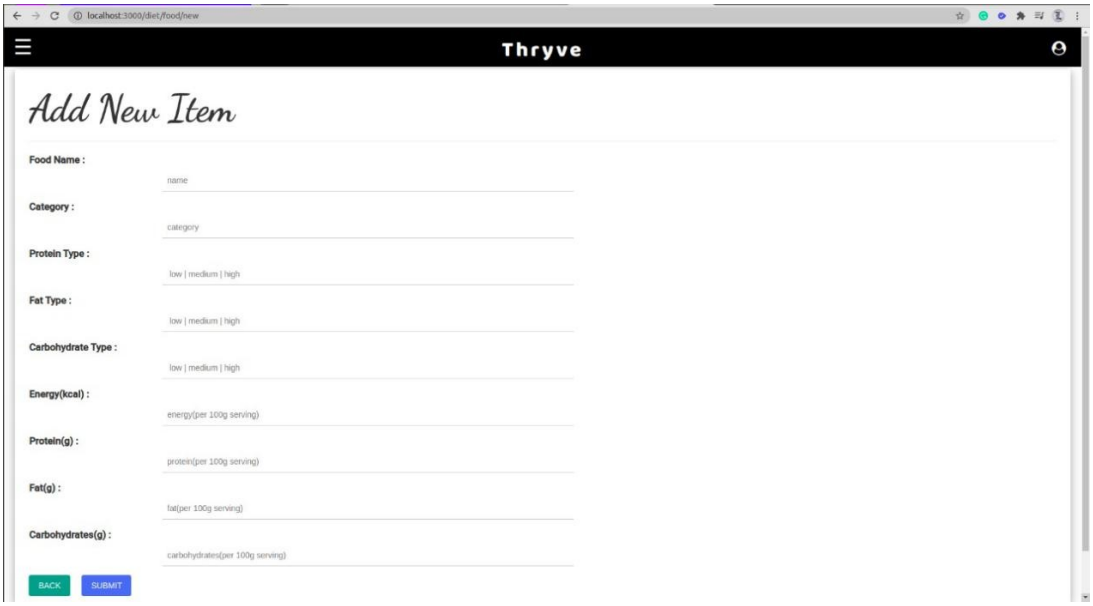
This page allows the user to create the diet that they will consume. Hovering over a dish gives the nutritional data of the dish. The quantity consumed of the dish can be selected before adding the dish to diet. Search functionality allows users to find dishes. New dishes can be added to the catalogue. These dishes will also receive a verified tag once an admin verifies it.



The checkout list keeps the record of the custom diet of the person and make an average count of the calories when the items are consumed.

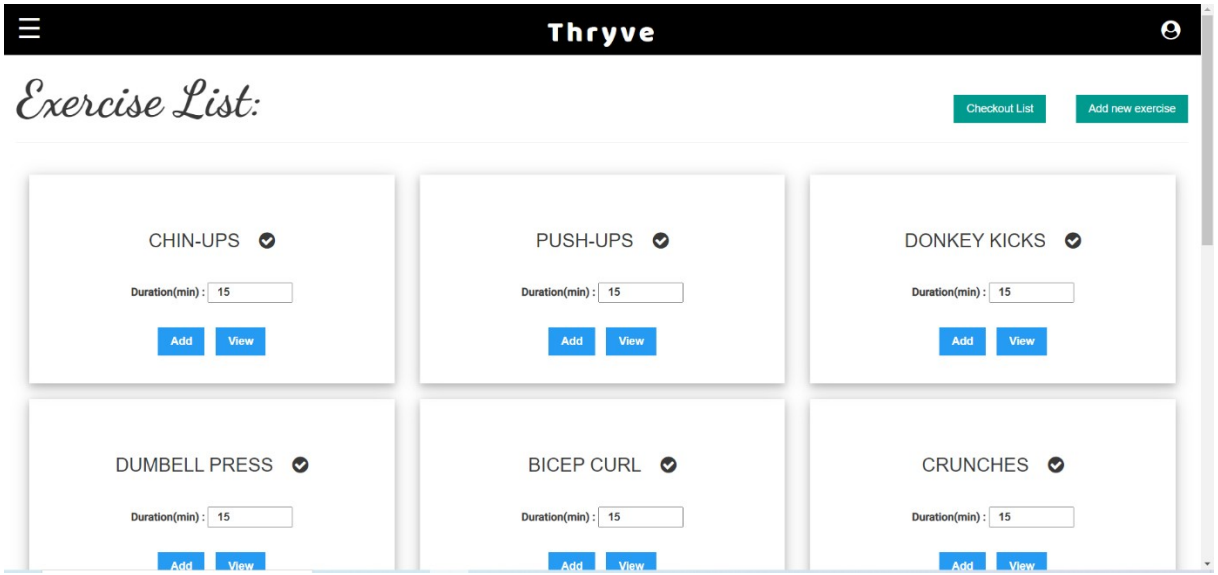


The dishes or fruits are categorized or classified into sub levels which makes it easy to search or find the particular dish or item easily.

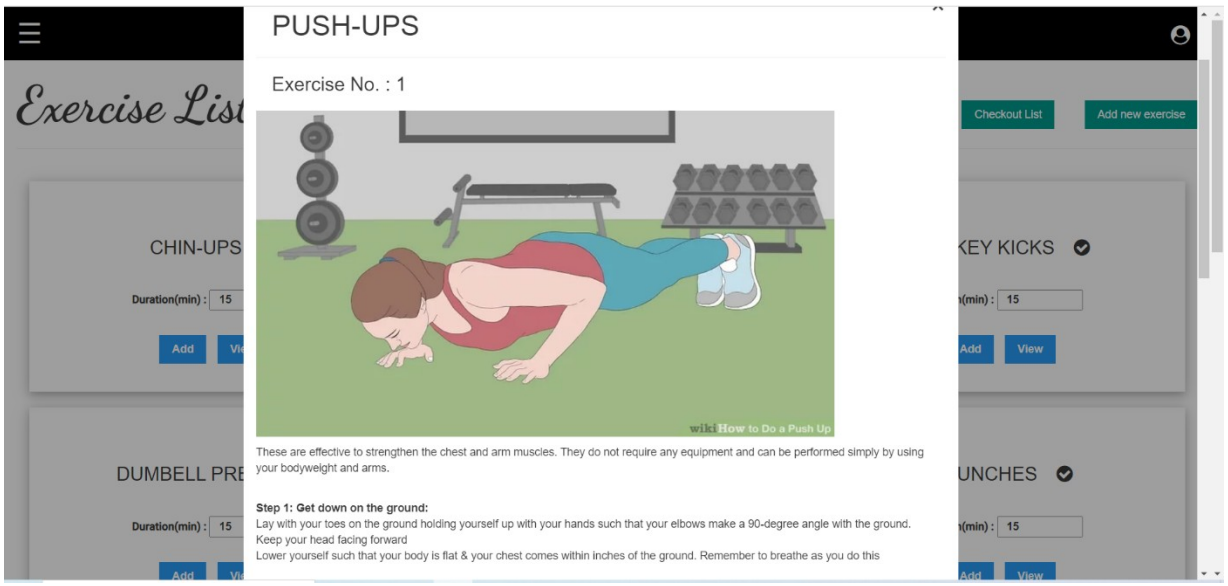


This page is used to add new items to the food catalogue. The added item is then approved or verified by a specific person.

1.3.5 Workout Builder

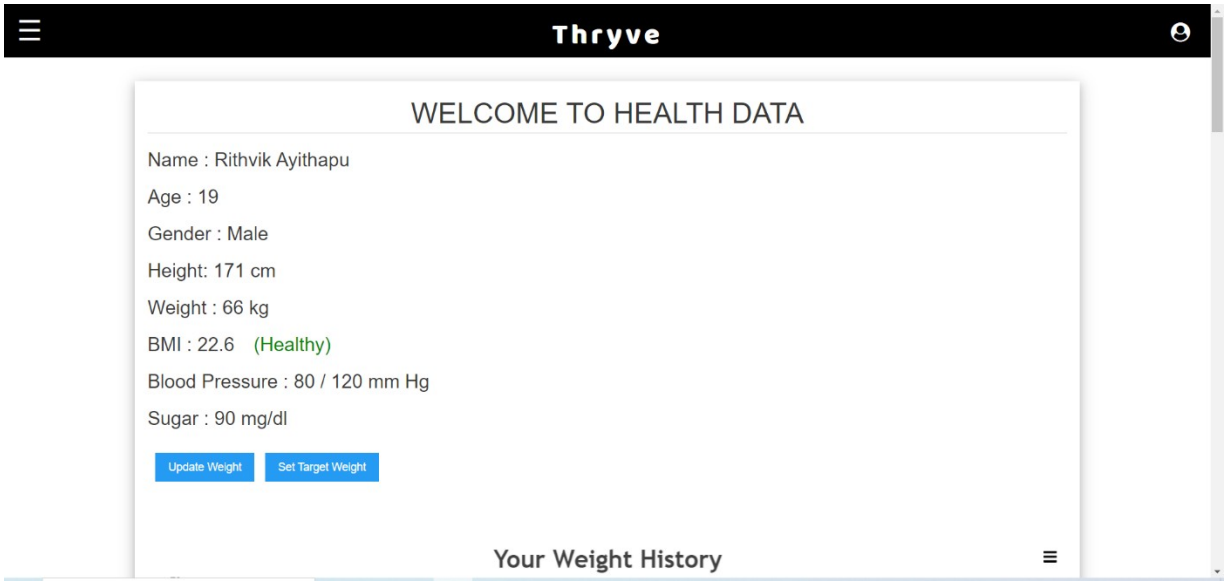


The exercise page allows users to choose exercises to make their workout routine. The duration of the exercise can be selected before adding the exercise to the workout. Based on the duration of the exercise the calories burnt will be calculated. New exercises can be added and the admin can verify them.

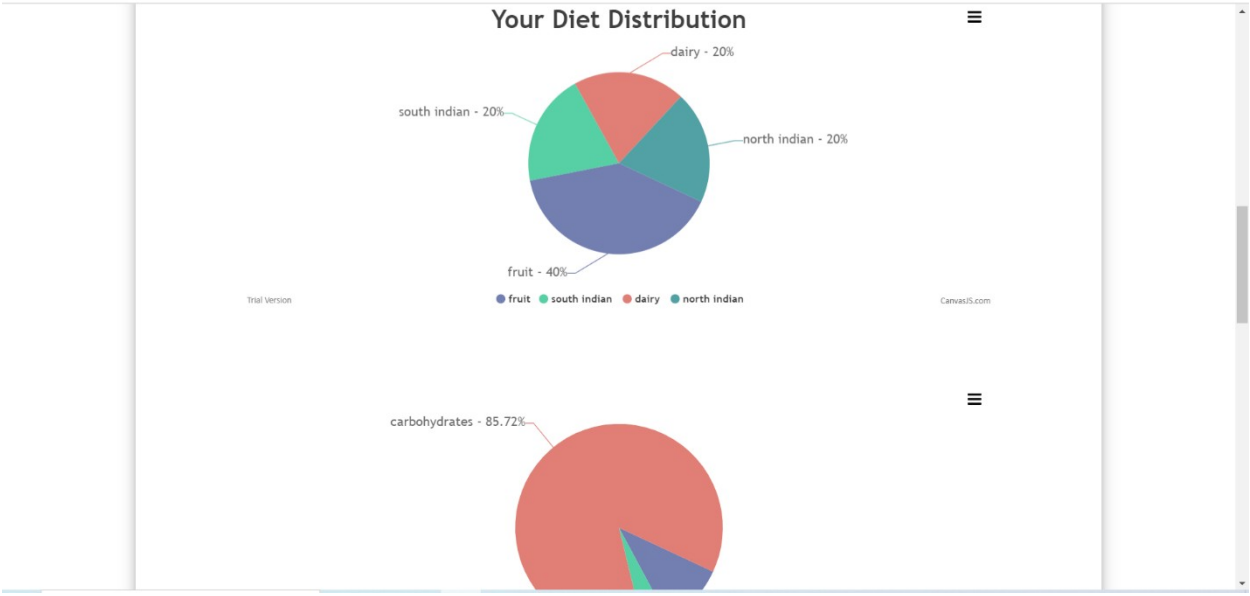


The “View” option in the exercise page teaches the user, how the particular exercise is performed.

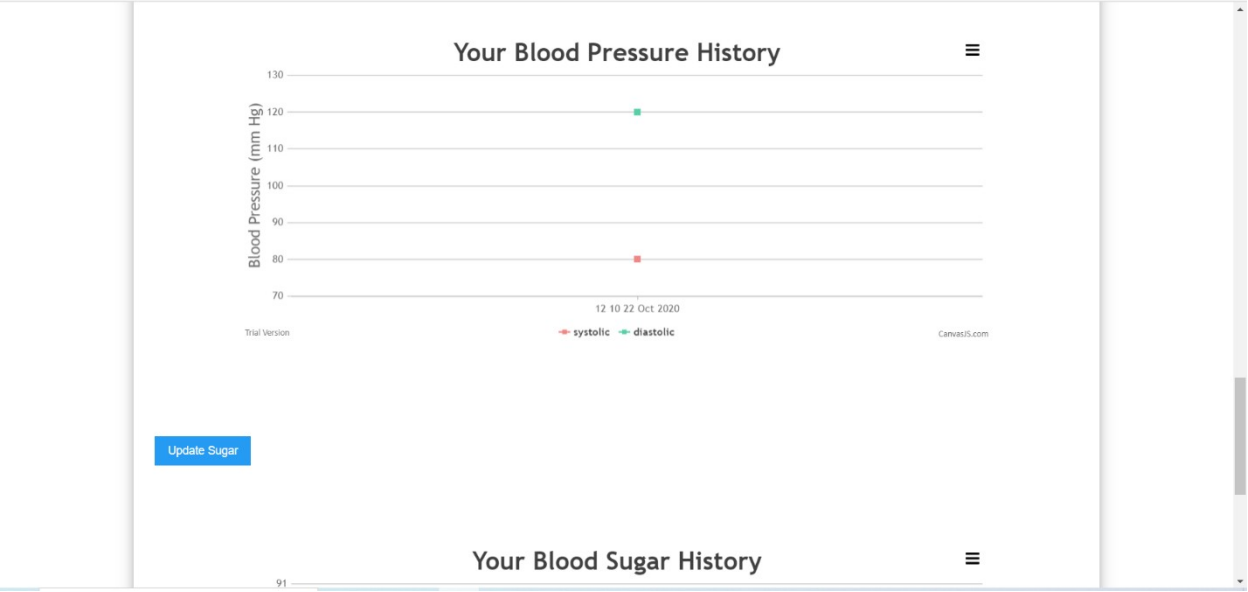
1.3.6 Health Info



The health info provides all the health information about the user in one page. At the top of the page the user can see their name, age, gender, height, weight, BMI, blood pressure and blood sugar. Following this the user's weight history is shown along with a graph plot showing the trends in the user's weight.

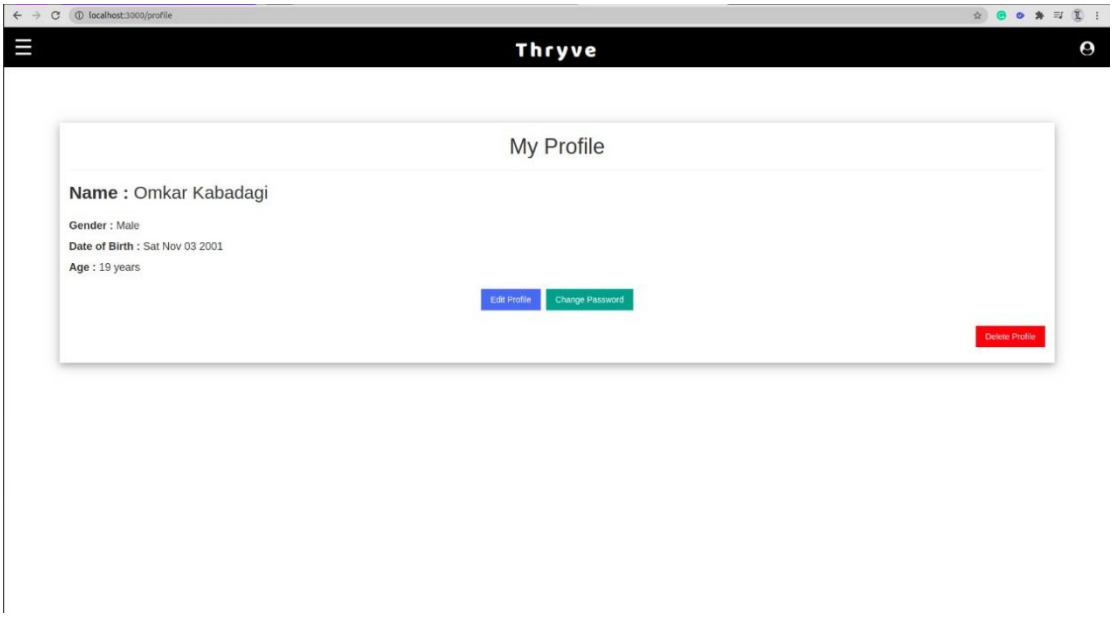


A diet distribution is also provided for the user to understand all the nutrients in the diet he/she is consuming.



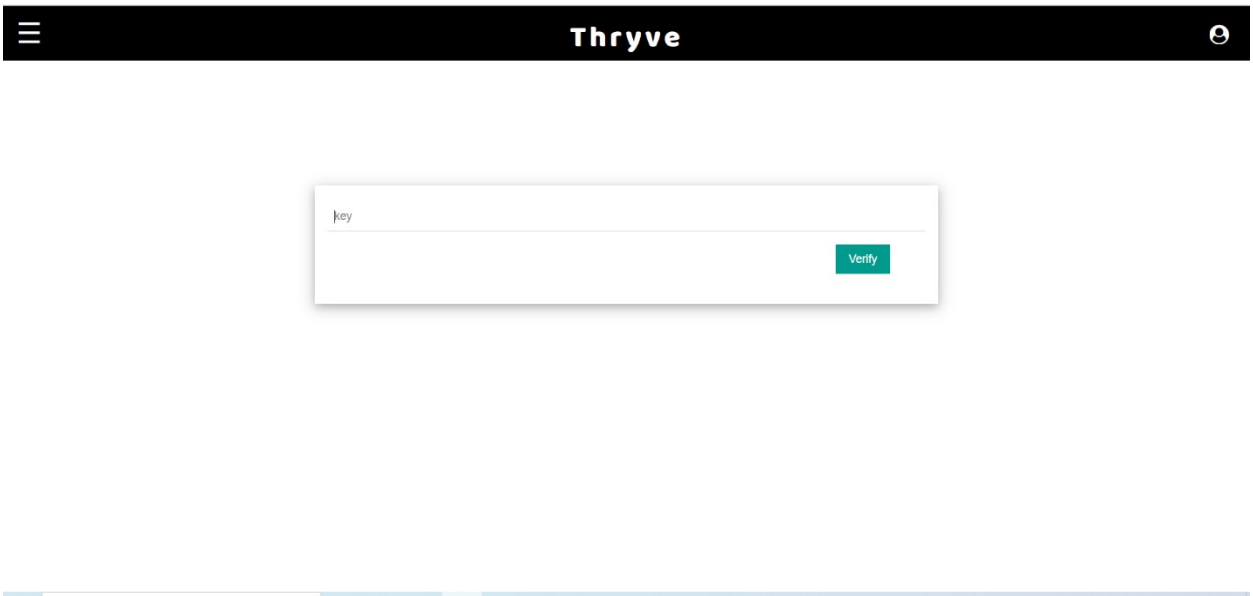
The blood sugar and blood pressure history are given in the form of a graph so that the user can see the trends in the respective category.

1.3.7 Profile



The user profile page shows the basic details of the user and enables the user to update or delete the profile. This page also enables the user to change his/her password.

1.3.8 Admin



The admin page asks the user to enter a key to change the status from user to admin. Once the key is entered the user shall be given all admin privileges like verifying dishes and exercises.

2. Testing

2.1 Test Case Generation

TEST CASE ID	TEST OBJECTIVE	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
TEST CASE 1	SIGN UP	(not registered) Username: omkar Email: omkar.kabadagi2019@vitstudent.ac.in Password: omkar123 Password confirm: omkar123	Msg: complete the verification /opt	Msg: complete the verification /opt	pass
TEST CASE 2	SIGN UP	(not registered) Username: omkar Email: omkar.kabadagi2019@vitstudent.ac.in Password: omkar Password confirm: omkar	Err: Minimum password length 8	Err: Minimum password length 8	pass
TEST CASE 3	SIGN UP	(not registered) Username: omkar Email: omkar.kabadagi2019@vitstudent.ac.in Password: omkar123 Password confirm: omkar1234	Err: password mismatch	Err: password mismatch	pass
TEST CASE 4	SIGN UP	(registered already) Username: omkar2 Email: omkar.kabadagi2019@vitstudent.ac.in Password: omkar123 Password confirm: omkar123	Err: account with this email already exist	Err: account with this email already exist	pass
TEST CASE 5	LOGIN	(registered already) Username: omkar Password: omkar123	Msg: Successful login /home	Msg: Successful login /home	pass

TEST CASE 6	LOGIN	(not registered) Username: omkar3 Password: omkar123	Err: incorrect username or password	Err: incorrect username or password	pass
TEST CASE 7	SEND OTP	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Click SEND OTP	Msg: Check your email for further instructions /otpCheck Mail received if email exists	Msg: Check your email for further instructions /otpCheck Mail received	pass
TEST CASE 8	ENTER OTP	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Enter OTP received in mail (within 1 hr of receiving mail)	Msg: Verified /profile/new	Msg: Verified /profile/new	pass
TEST CASE 9	ENTER OTP	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Enter OTP received in mail (after 1 hr of receiving mail)	Msg: otp expired /otp	Msg: otp expired /otp	pass
TEST CASE 10	ENTER OTP	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Entered incorrect OTP	Msg: incorrect otp /otp	Msg: incorrect otp /otp	pass
TEST CASE 11	FORGOT PASSWORD	(email is registered) Email: omkar.kabadagi2019@vitstudent.ac.in	Msg: email has been sent to mail with further details	Msg: email has been sent to mail with further details	pass

			Generate temporary link	Generate temporary link	
TEST CASE 12	FORGOT PASSWORD	(email is not registered with the site) Email: okabadagi@live.co.uk	Err: no account with given mail found	Err: no account with given mail found	pass
TEST CASE 13	RESET PASSWORD	(email: omkar.kabadagi2019@vitstudent.ac.in) Click link in mail	/ reset/:token	/ reset/:token	pass
TEST CASE 14	RESET PASSWORD	(email: omkar.kabadagi2019@vitstudent.ac.in) New password: omkar123	/home Password changed	/home Password changed	pass
TEST CASE 15	NEW PROFILE AFTER OTP VERIFICATION	(registered in with omkar.kabadagi2019@vitstudent.ac.in) First Name: Omkar Last Name: Kabadagi Gender: Male DOB: 3/11/2001 (doesn't accept future dates) Height: 168 cm Weight: 65 kg	/home	/home	pass
TEST CASE 16	PROFILE EDIT	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Edited last name to K from Kabadagi	Profile Updated /home	Profile Updated /home	pass
TESTCASE 17	ADMIN ROLE	(registered in with omkar.kabadagi2019@vitstudent.ac.in)	/home Admin rights	/home Admin rights	pass

		Key: abcxyz@123 (correct key)	given	available	
TESTCASE 18	ADMIN ROLE	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Key: acbdefg (wrong key)	/home Msg: Incorrect key Admin rights denied	/home Msg: Incorrect key Admin rights denied	pass
TEST CASE 19	HOME	After login or registration	Dashboard Trending dishes and exercises Diet and Exercise List	Dashboard Trending dishes and exercises Diet and Exercise List	pass
TEST CASE 20	MATCH DASHBOARD WITH DIET AND WORKOUT LIST	Match values of Diet and Workout list with Dashboard	Must Match	Matched	pass
TEST CASE 21	UPDATE CHECK LIST OF DIET	(apple and mango in checklist) Check Apple	Apple checked Dashboard Updated	Apple checked Dashboard Updated	pass
TEST CASE 22	UPDATE CHECK LIST OF WORKOUT	(burpees and push ups in checklist) Check burpees	Burpees checked Dashboard Updated	Burpees checked Dashboard Updated	pass
TEST CASE 23	DIET PAGE	Diet page clicked	/diet Display the catalogue of foods	/diet Display the catalogue of foods	pass

TEST CASE 24	DIET PAGE AUTOCOMPLETE SEARCH	Input: pla	List of dishes starting with ‘pla’	Plain dosa	pass
TEST CASE 25	DIET PAGE FILTER	Category: Bread Protein: Medium Fat: Low Carbohydrates: High	Filter breads with mentioned nutrient composition	Breads with medium protein, low fat and high carbohydrates	pass
TEST CASE 26	DIET PAGE ADD DISHES	Add apple, and mango to checkout cart with default quantity	100g of Apple and Mango in Checkout list	100g of Apple and Mango in Checkout list	pass
TEST CASE 27	DIET PAGE CHANGE DISH QUANTITY	Change quantity of mango to 500g	100g of Apple and 500g Mango in Checkout list	100g of Apple and 500g Mango in Checkout list	pass
TEST CASE 28	DIET PAGE REMOVE DISH	Remove apple from checkout list	500g Mango in Checkout list	500g Mango in Checkout list	pass
TEST CASE 29	ADD CUSTOM DISH	Name: Banana and Papaya Smoothie Category: Juice Protein: Medium (0.8 g) Fat: Low (0.2 g) Carbohyrate: High (13.8 g)	Banana and Papaya Smoothie in database under juices	Banana and Papaya Smoothie in database under juices	pass
TEST CASE 30	VERIFY CUSTOM DISHES OF USERS BY ADMIN	Verify: Banana and Papaya Smoothie	Banana and Papaya Smoothie verified with tick	Banana and Papaya Smoothie verified with tick	pass

TEST CASE 31	WORKOUT BUILDER PAGE	Workout builder page clicked	/workoutbuilder Display the exercises	/workoutbuilder Display the exercises	pass
TEST CASE 32	WORKOUT BUILDER PAGE ADD EXERCISES	Add burpees with default time	15 min of Burpees in Checkout list	15 min of Burpees in Checkout list	pass
TEST CASE 33	WORKOUT BUILDER CHANGE EXERCISE TIME	Change time of burpees to 10 min	10 min of Burpees in Checkout list	10 min of Burpees in Checkout list	pass
TEST CASE 34	WORKOUT BUILDER REMOVE DISH	Burpees from checkout list	Empty Check out list	Empty Check out list	pass
TEST CASE 35	ADD CUSTOM EXERCISE	Name: My exercise MET: 3 Steps: These are the steps	My exercise added to database	My exercise added to database	pass
TEST CASE 36	VERIFY CUSTOM EXERCISE OF USERS BY ADMIN	Verify: My exercise	My exercise verified with tick	My exercise verified with tick	pass
TEST CASE 37	HEALTINFO	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Health info clicked	/healthinfo User health info display	/healthinfo User health info display	pass
TEST CASE 38	UPDATE WEIGHT	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Update weight to 63 kg	Update weight to 63 kg with new data point on weight	Update weight to 63 kg with new data point on weight	pass

			graph	graph	
TEST CASE 39	SET TARGET WEIGHT	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Set new target weight to 60 kg	Target weight to 60 kg, display in graph	Target weight to 60 kg, display in graph	pass
TEST CASE 40	UPDATE BLOOD PRESSURE	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Update BP to 120/80 mmHg	Update BP to 120/80 mmHg with new data point on BP graph	Update BP to 120/80 mmHg with new data point on BP graph	pass
TEST CASE 41	UPDATE BLOOD SUGAR	(registered in with omkar.kabadagi2019@vitstudent.ac.in) Update Sugar to 140 mg/dl	Update Sugar to 140 mg/dl with new data point on Sugar graph	Update Sugar to 140 mg/dl with new data point on Sugar graph	pass

2.2 Testing Tool

We have used CrossBrowserTesting by Smartbear as out testing tool to perform automated and performance testing.

2.3.1 Details of Testing Tool

CrossBrowserTesting allows users to easily run manual, visual and Selenium tests in the cloud and test on more than 2050 real desktop and mobile browsers. Here we have used the software for performance testing.

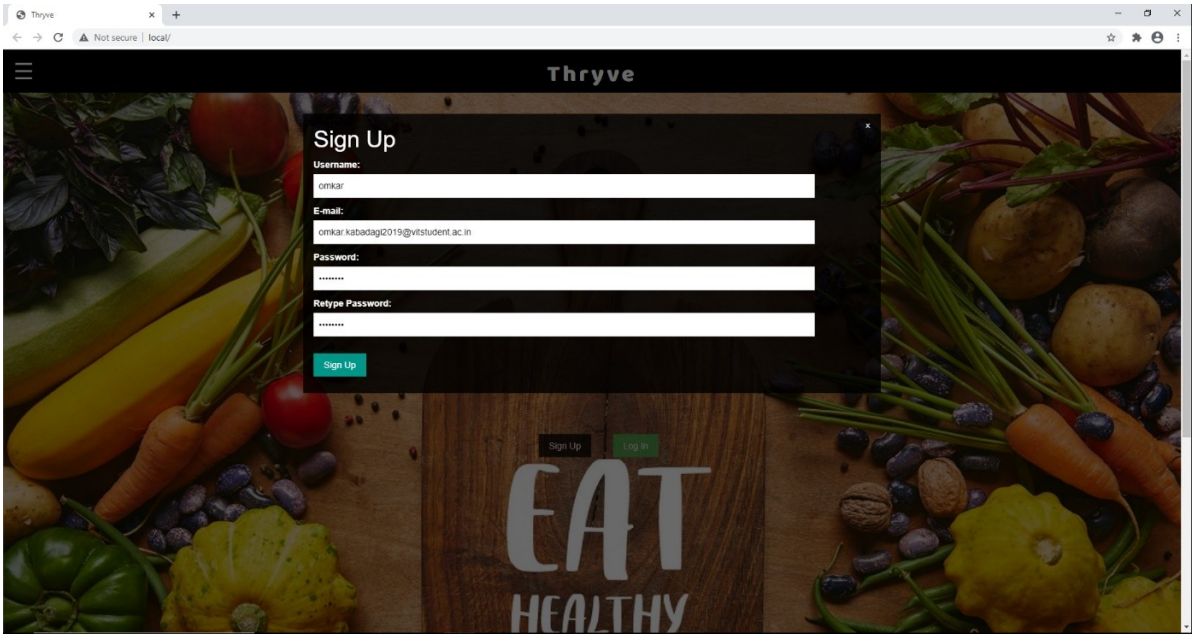
2.3.2 Configuration of Testing Tool

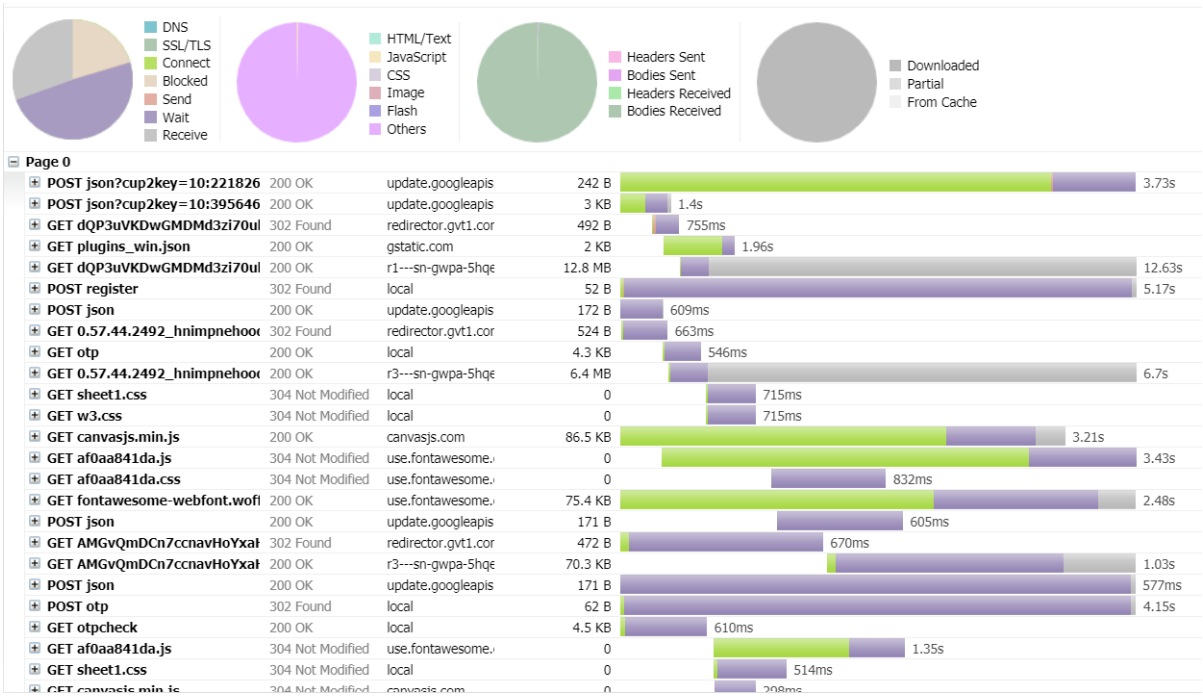
The testing tool is a licensed software. The free trial was used to run test cases. In terms of configuration used for testing the website the following steps were followed:

- Start MongoDB on Localhost.
- Start NodeJS on Localhost.
- Set the root directory of the project using the CrossBrowserTesting Command Line Interface.
- Start Live Test by remote SSH into CrossBrowserTesting with Desktop Setup
    - OS: Windows 10
    - Browser: Google Chrome 86 (64-bit)

2.3.3 Graph Generation with table

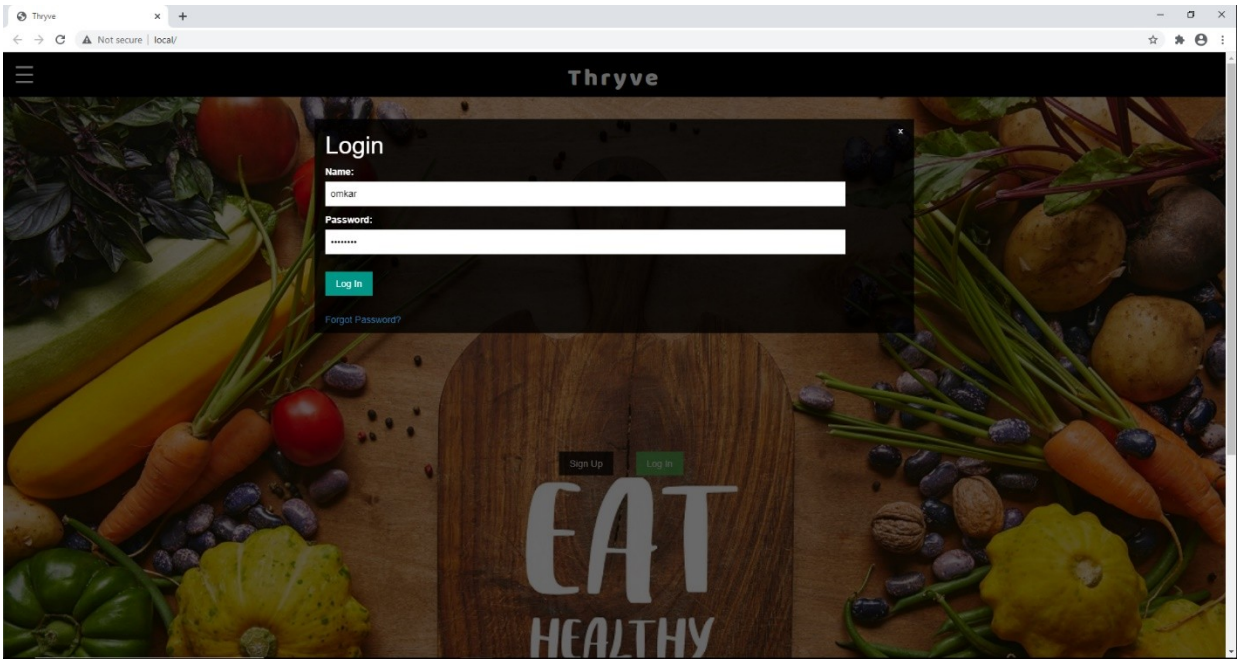
2.3.3.1 Sign Up

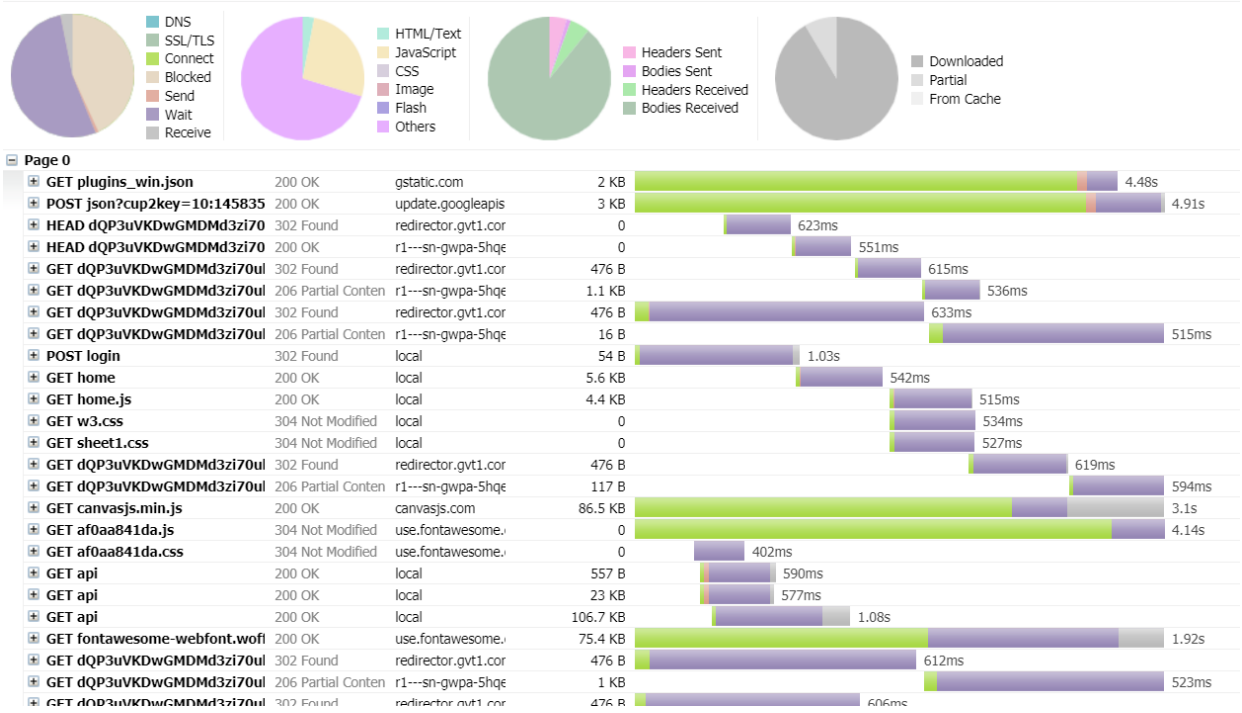




Link to test results: https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54574888/8cd60864

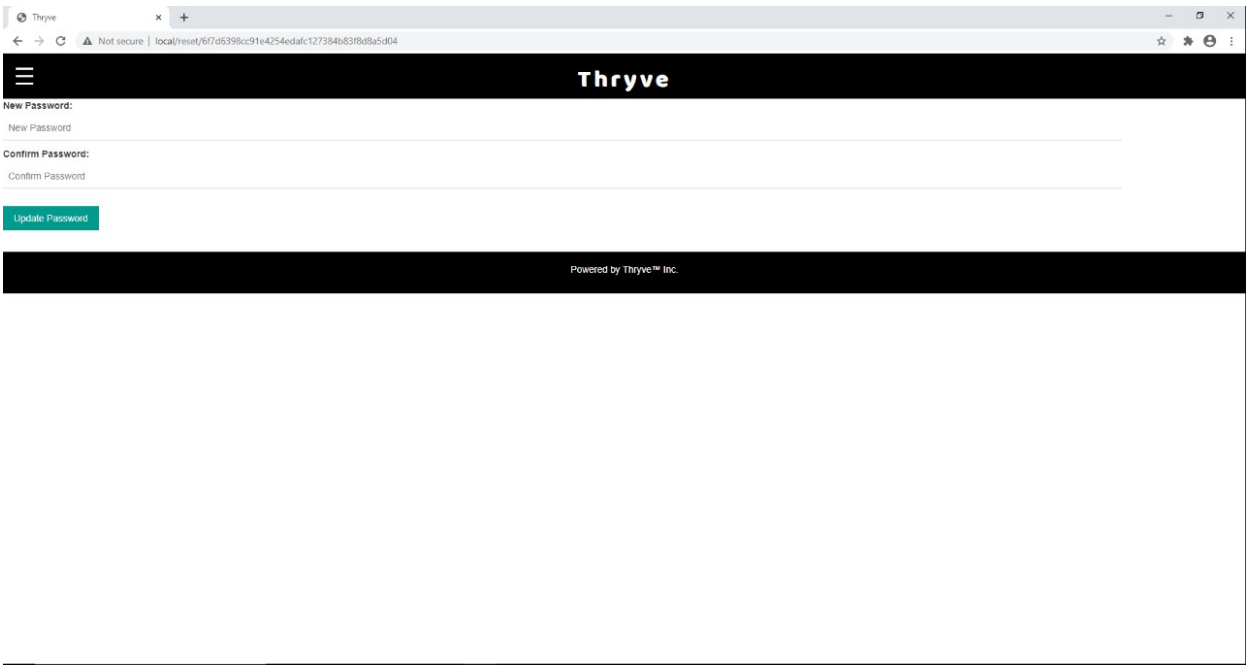
2.3.3.2 Login

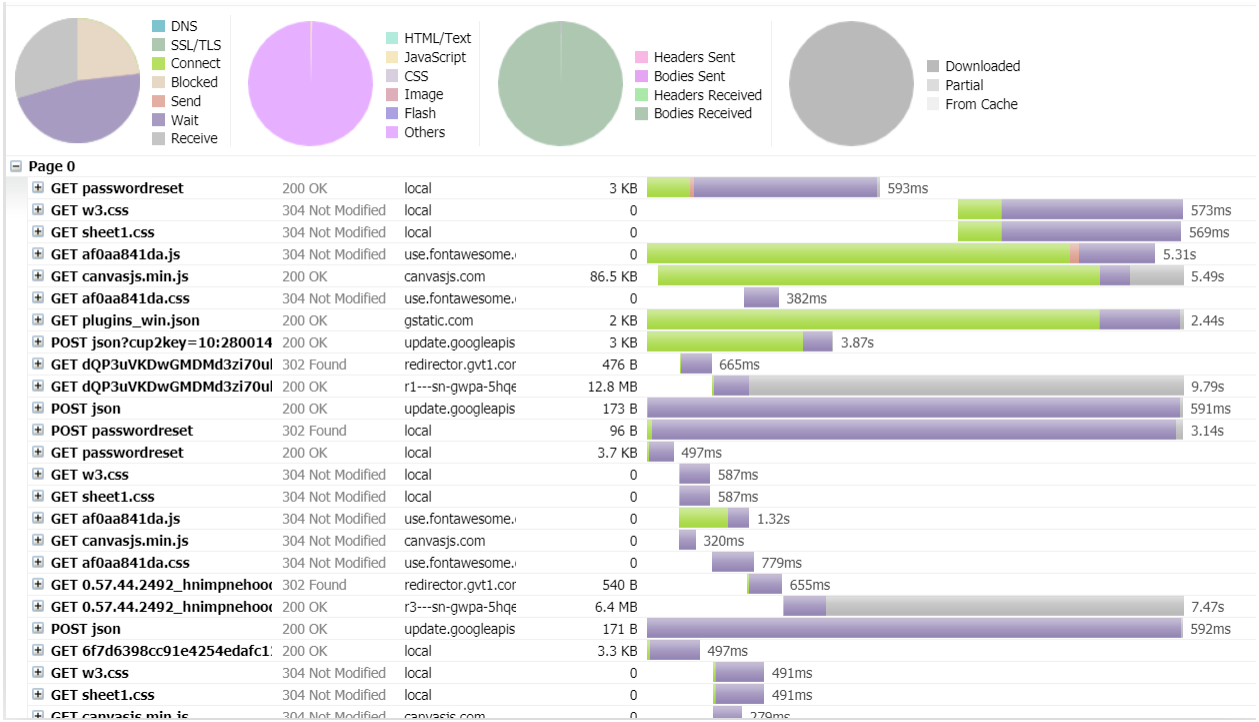




Link to test results: https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54575106/0caa3f98

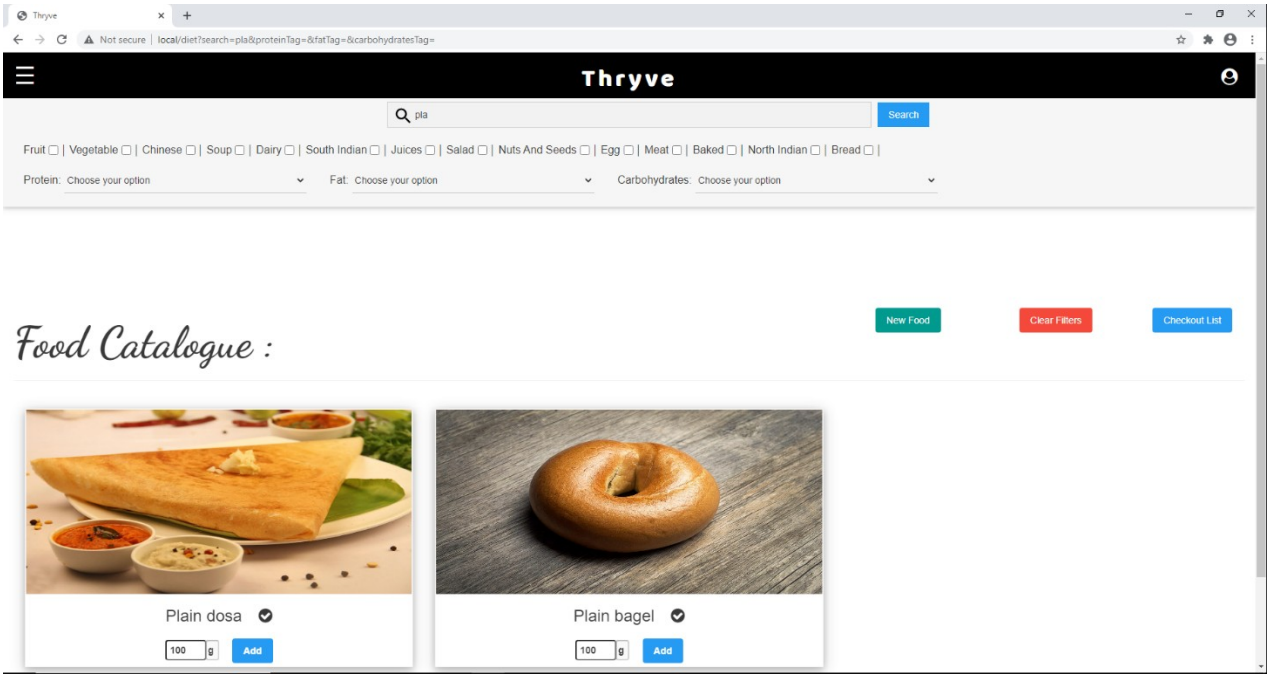
2.3.3.3 Password Reset

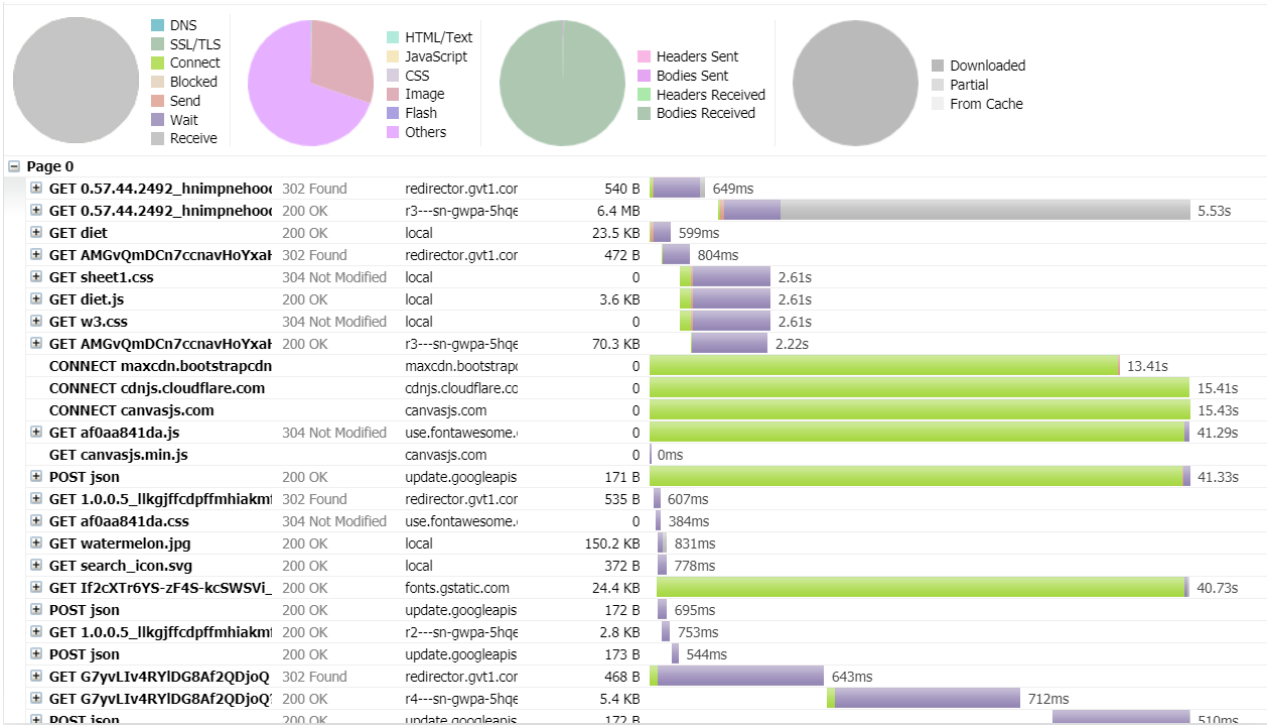




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54575939/bfe86176

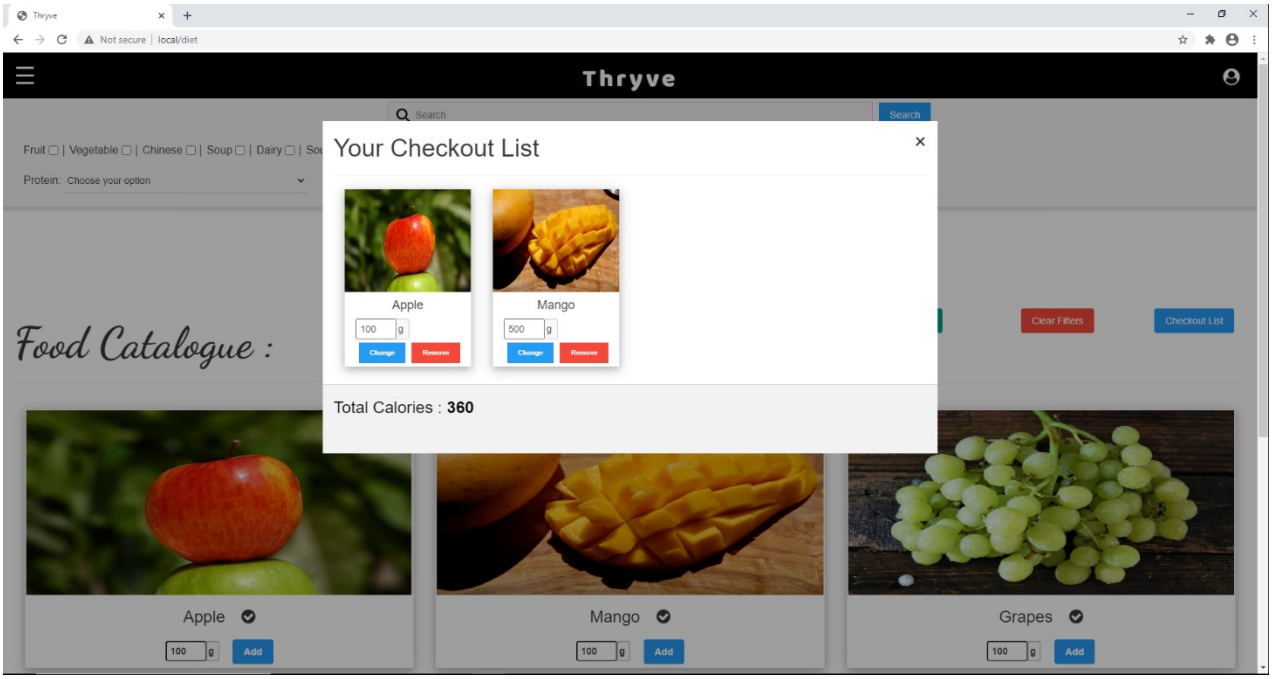
2.3.3.4 Diet Page, Autosearch, Filter

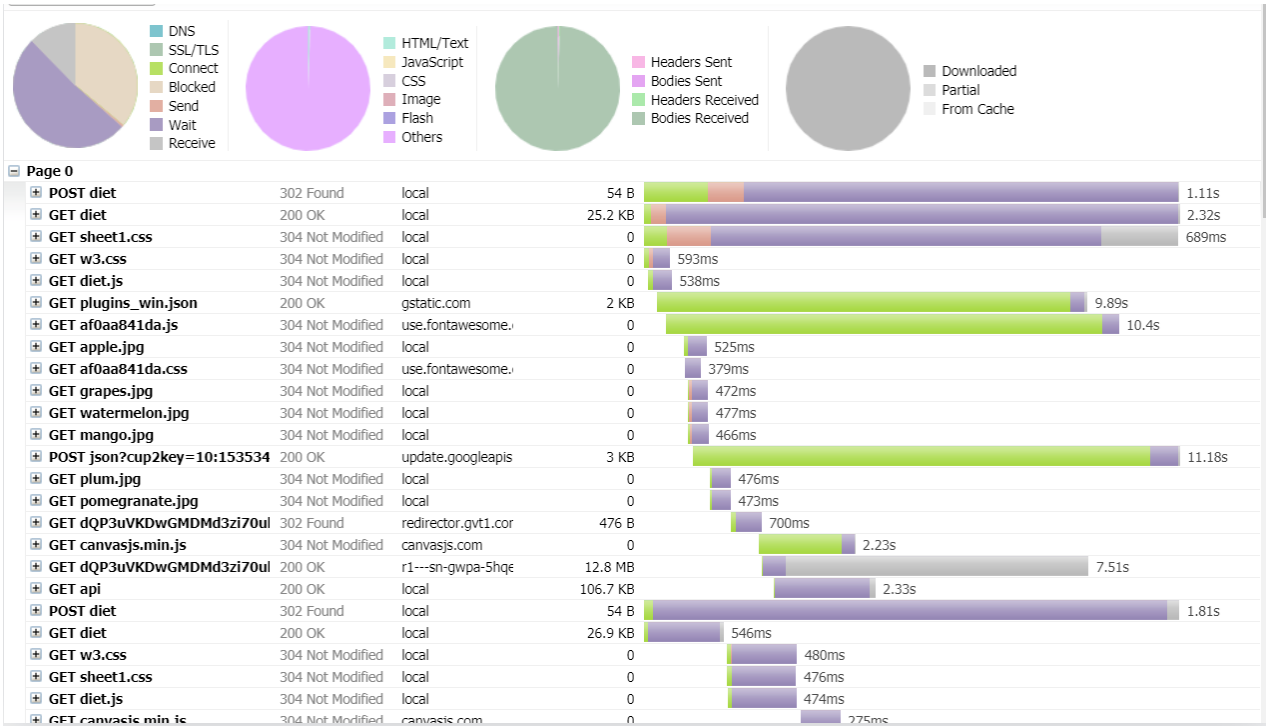




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54576122/8529f137

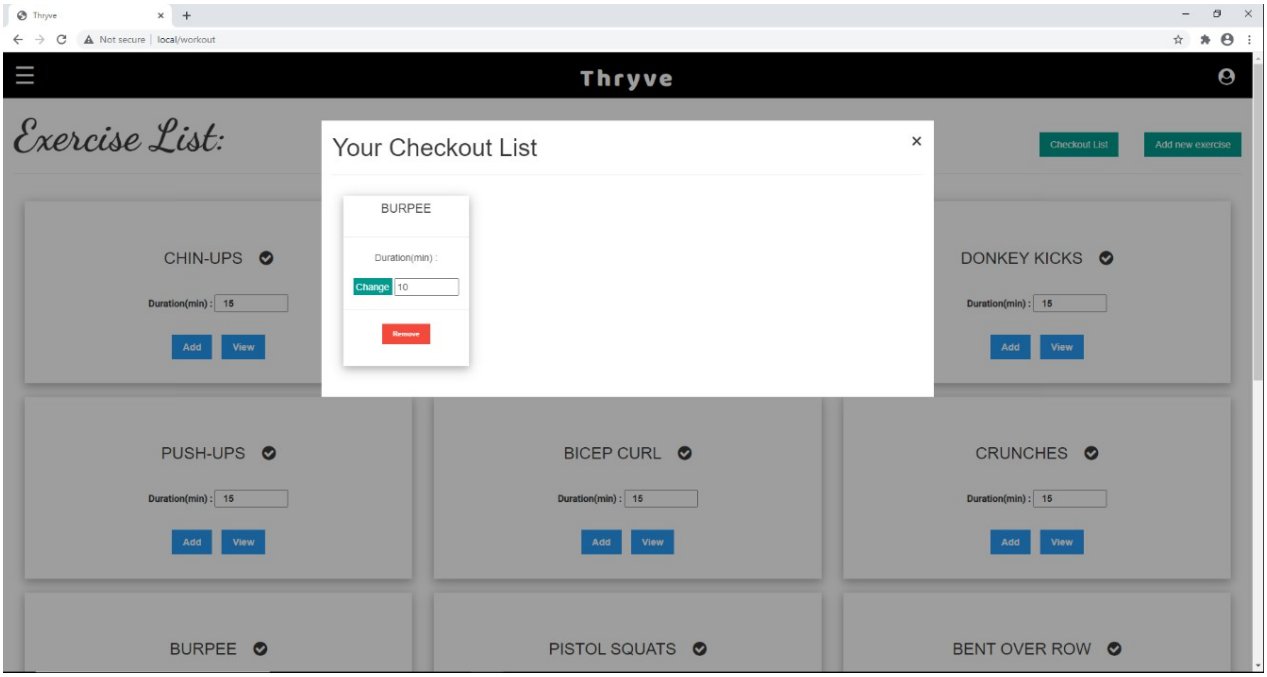
2.3.3.5 Diet Page Add, Change, Remove

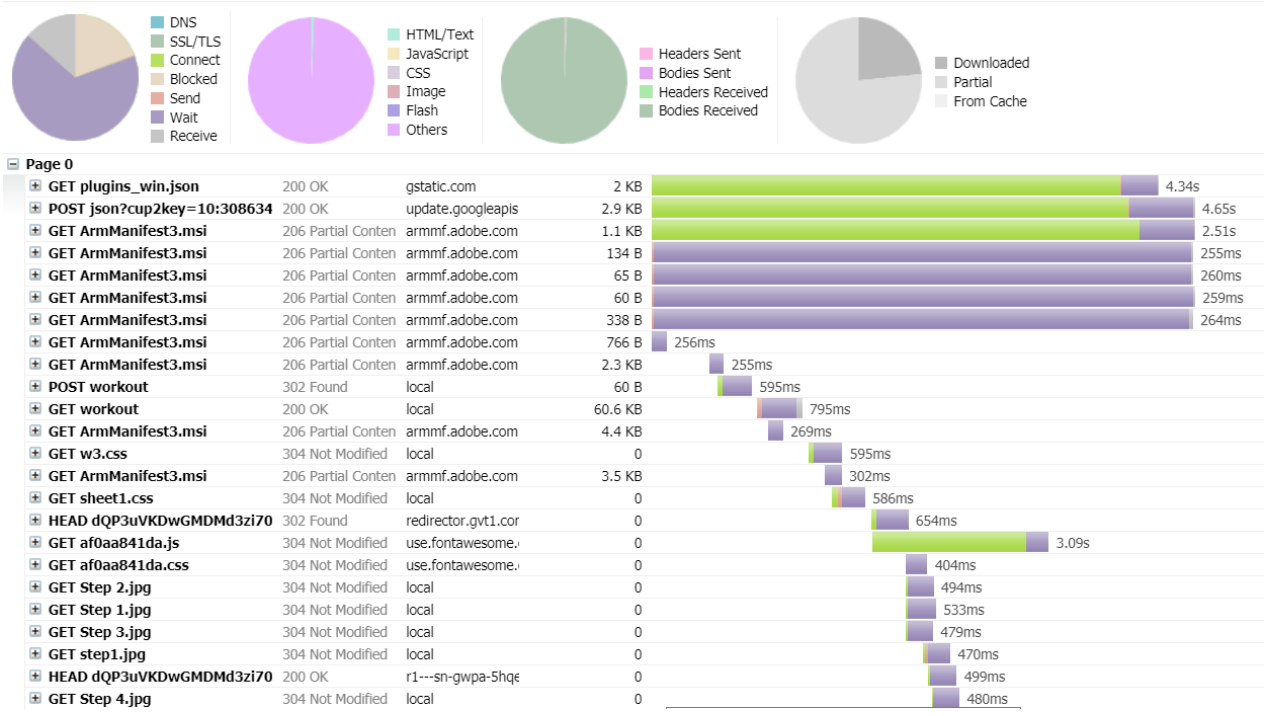




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54576297/69ac1e5a

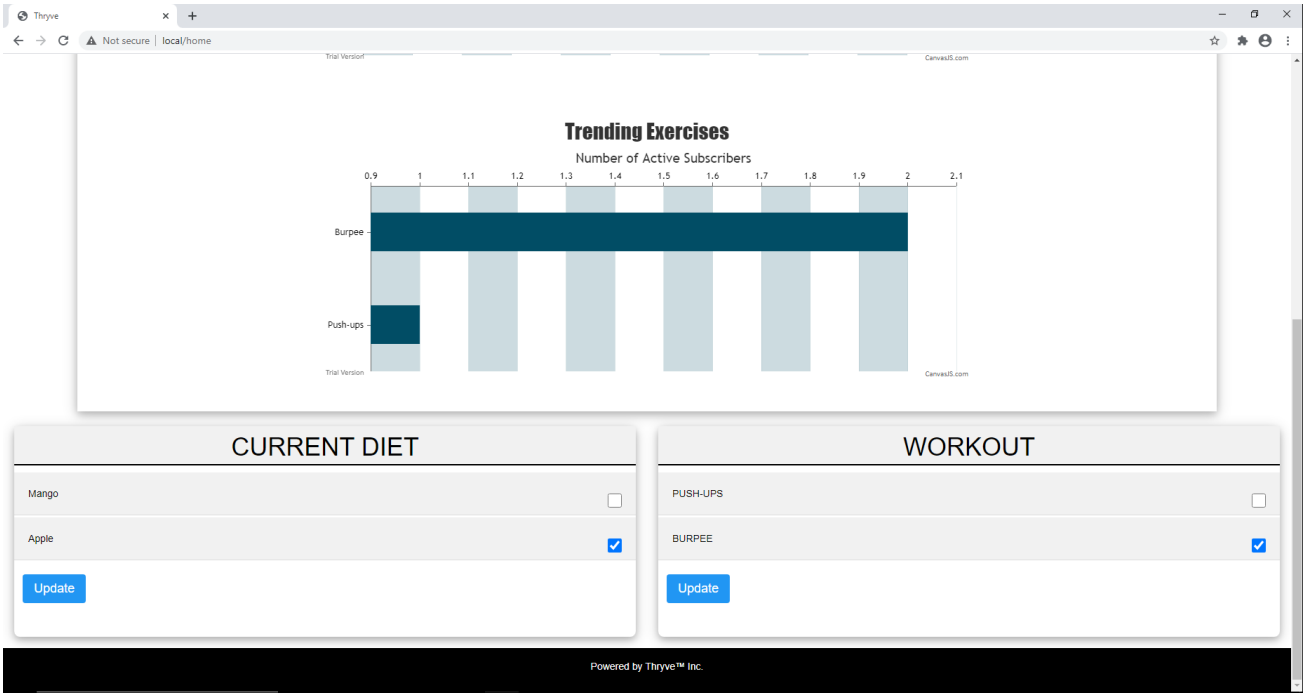
2.3.3.6 Workout Builder Page, Add, Change, Remove

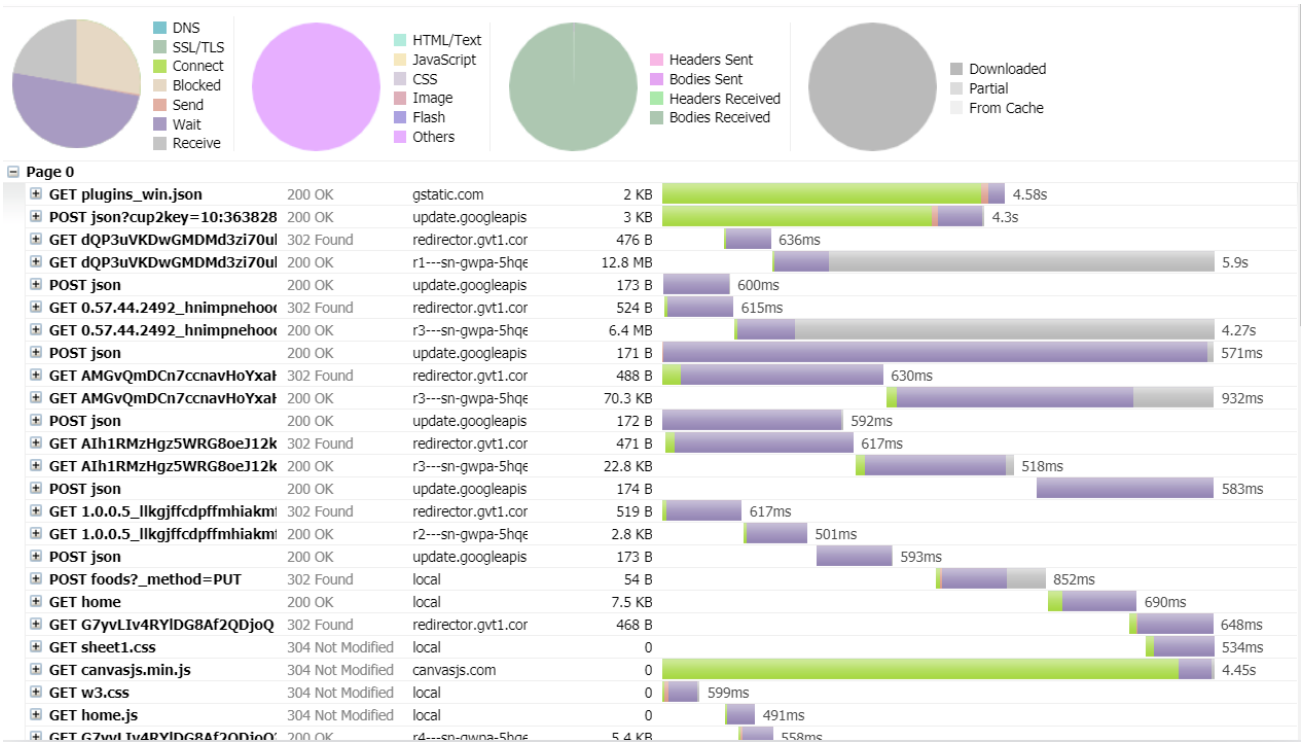




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54576688/887eeb45

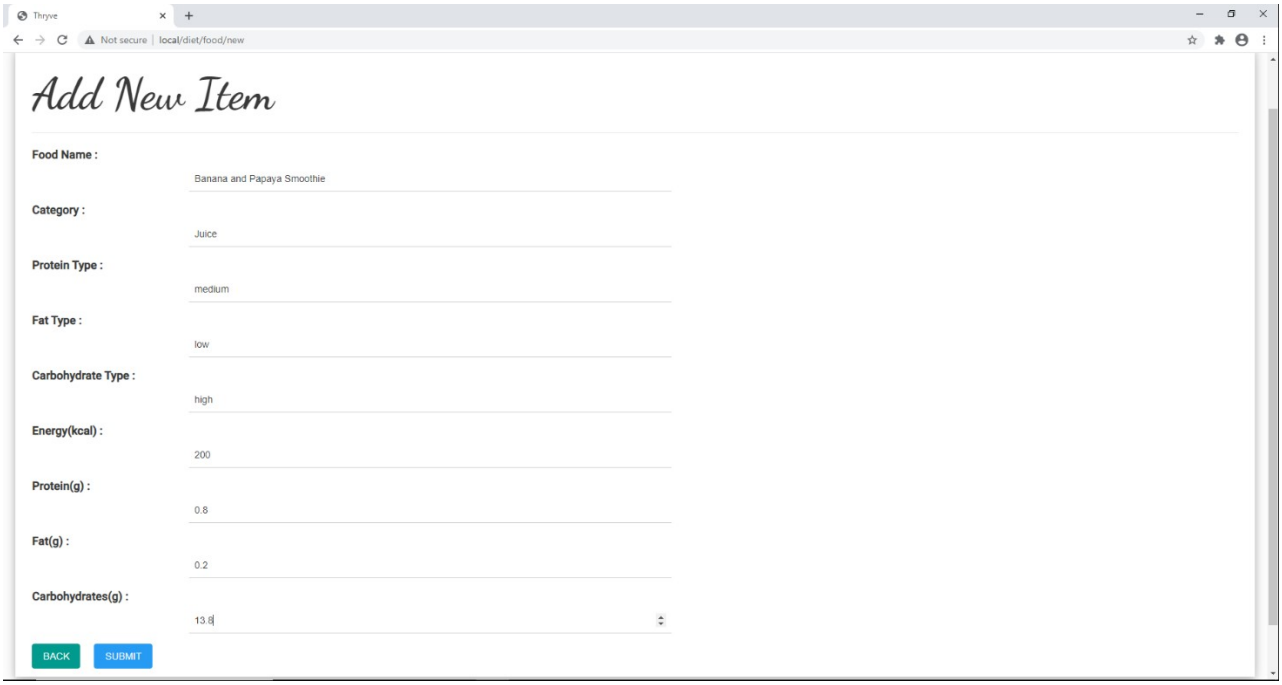
2.3.3.7 Dashboard, Trending and Checklist

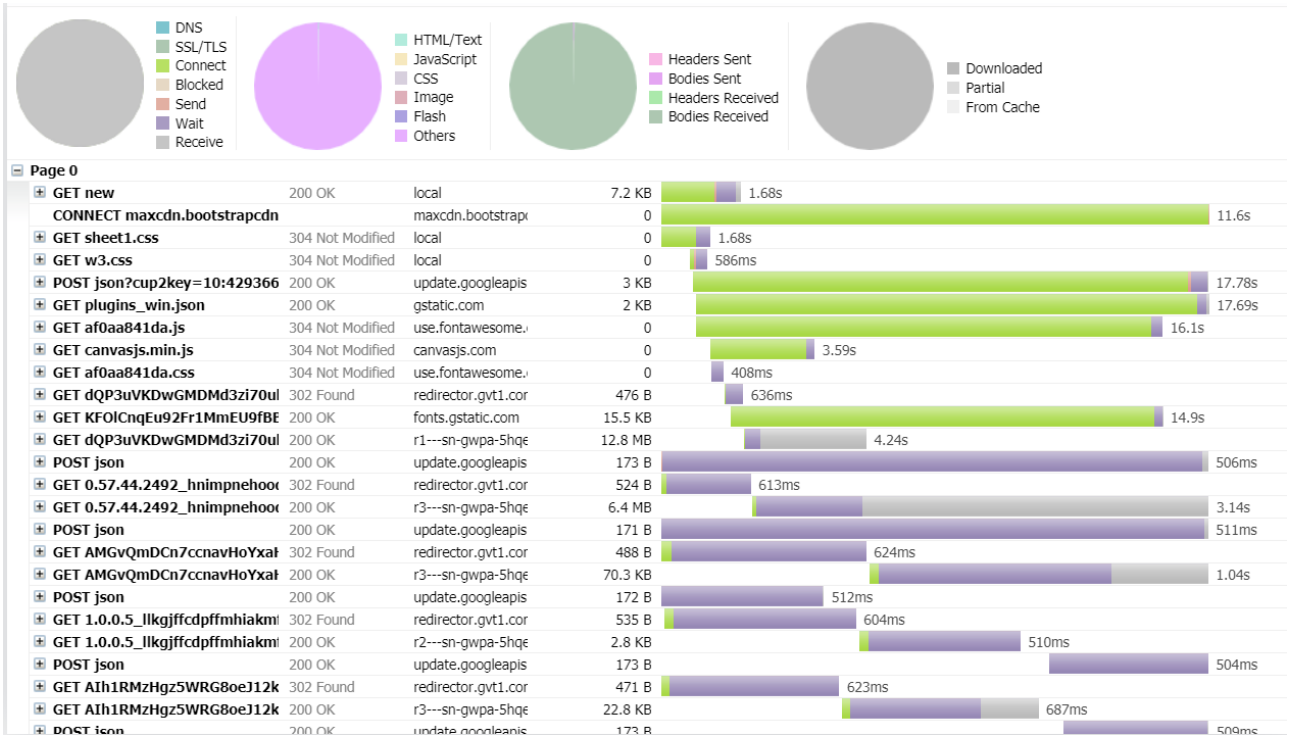




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54576551/46d5e0dc

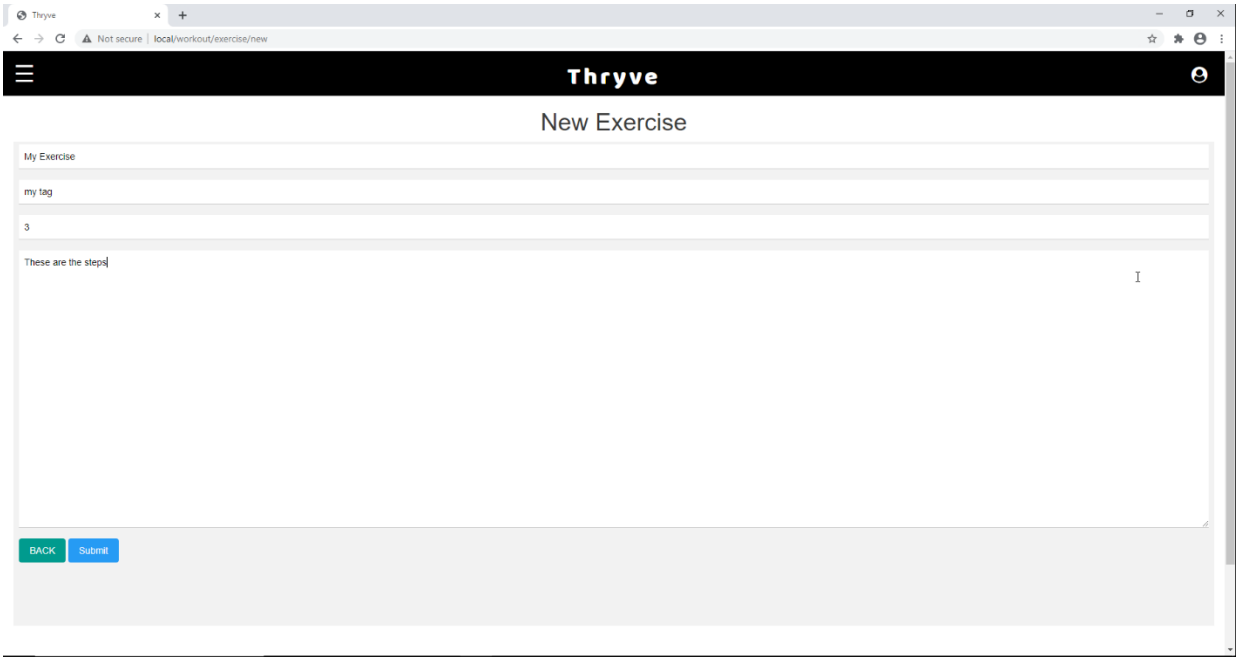
2.3.3.8 Add Custom Dish

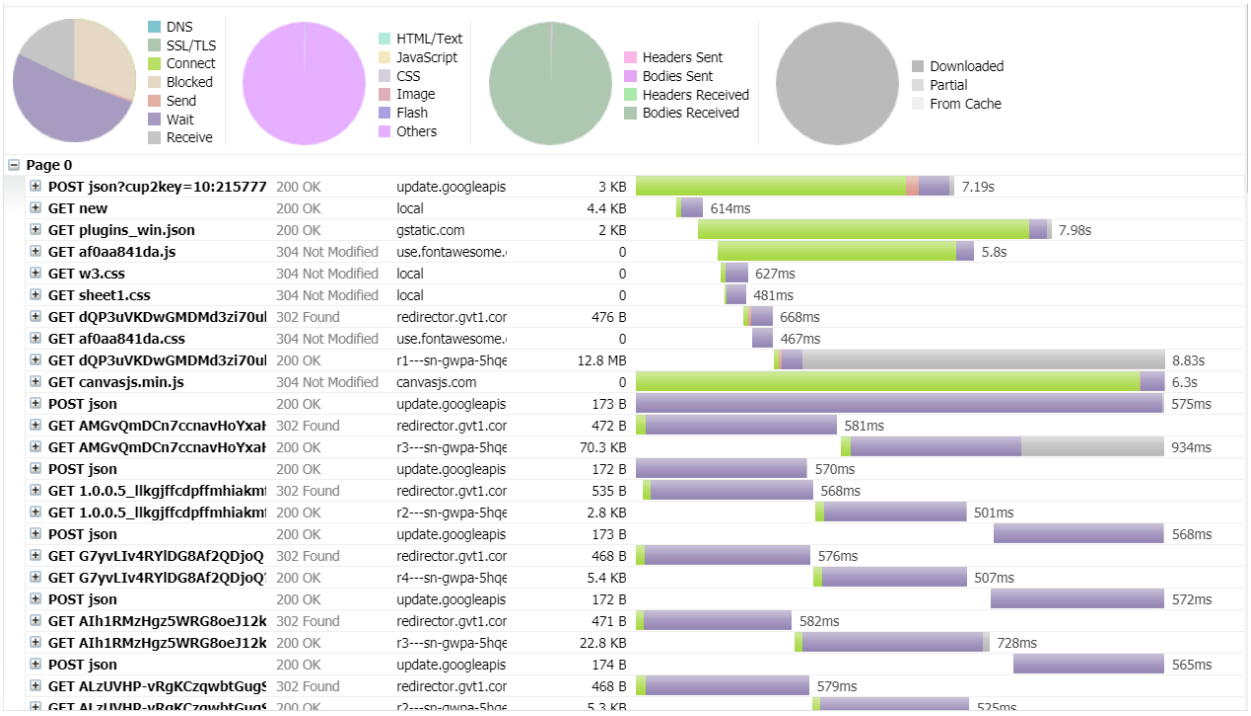




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54576806/91ae3e08

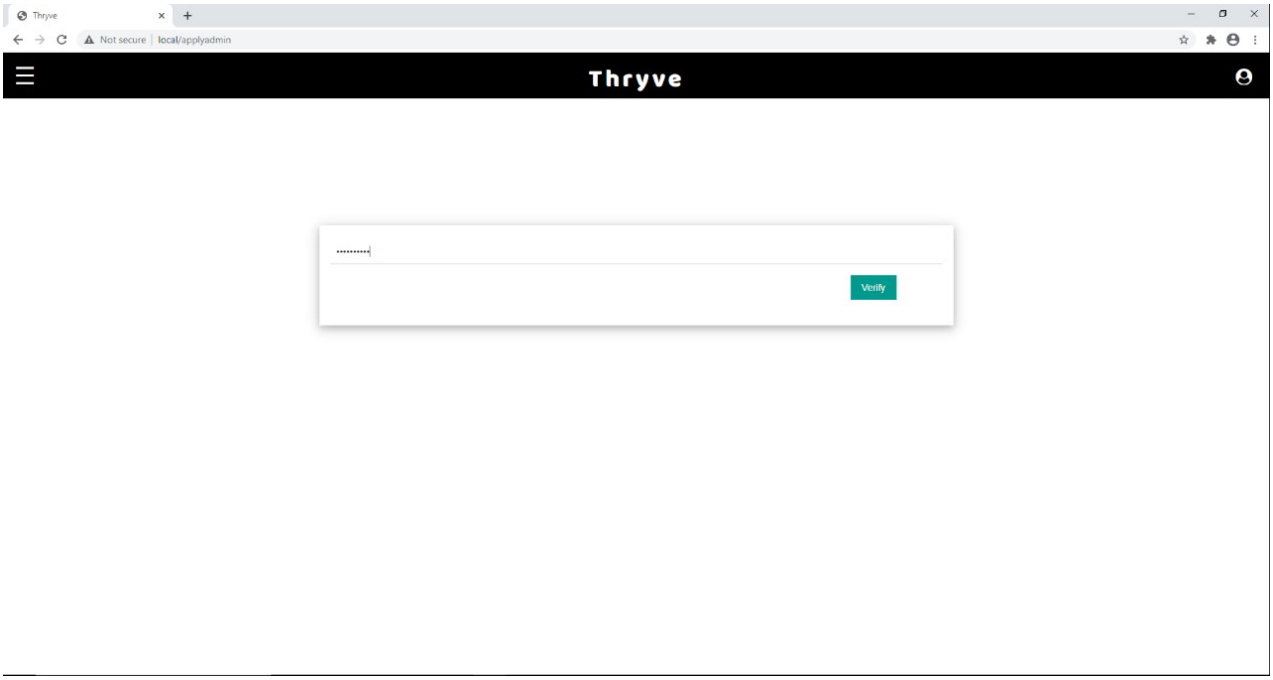
2.3.3.9 Add Custom Exercise

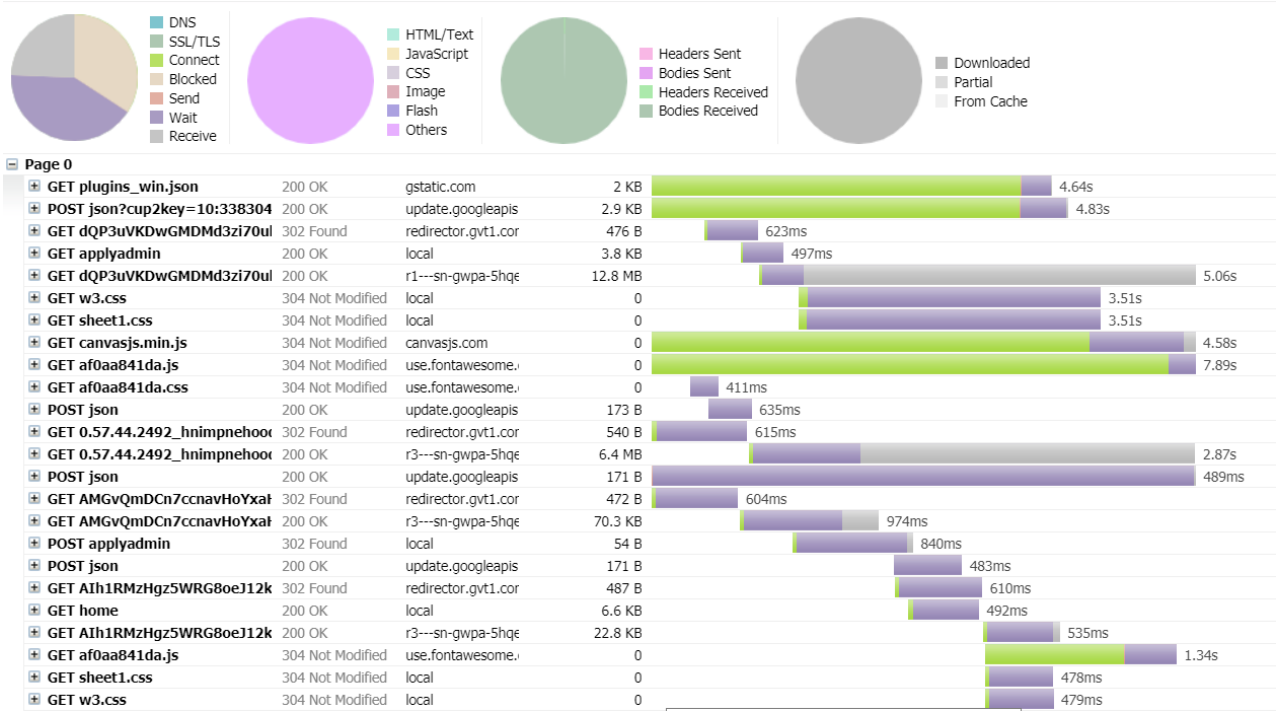




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54576934/96c64c89

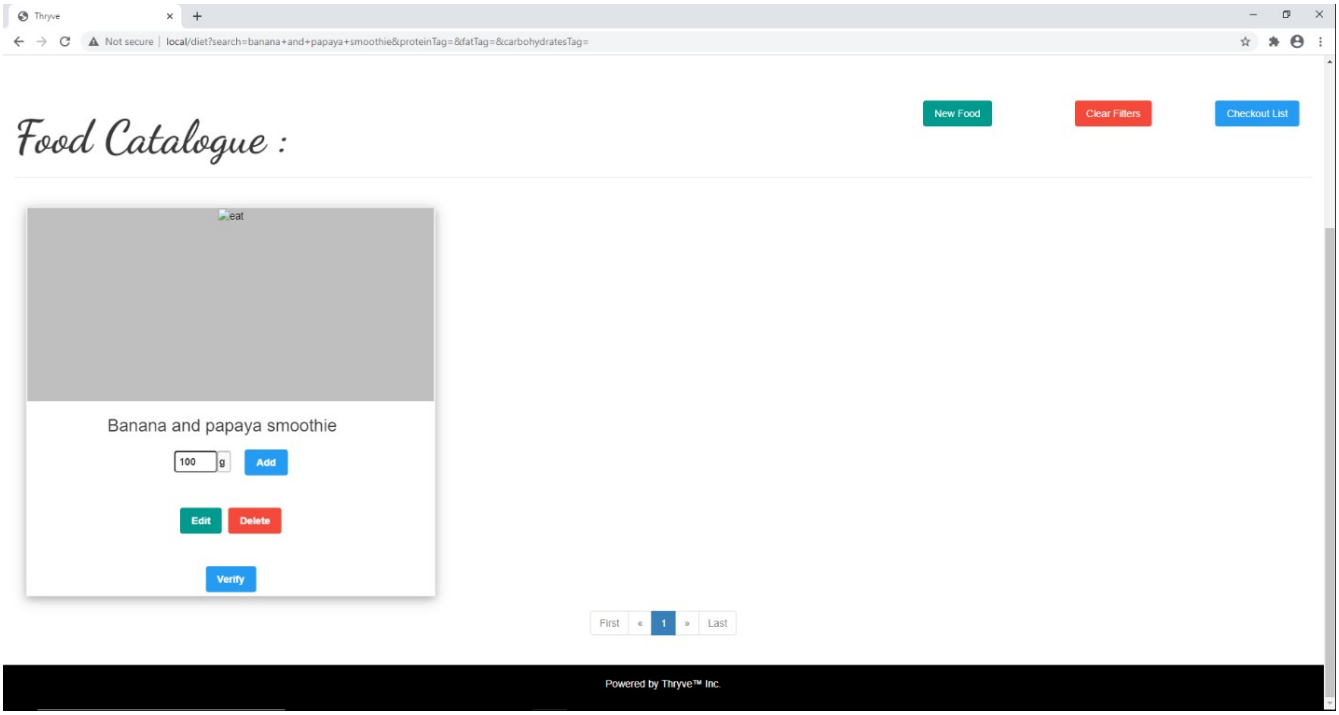
2.3.3.10 Make Admin User

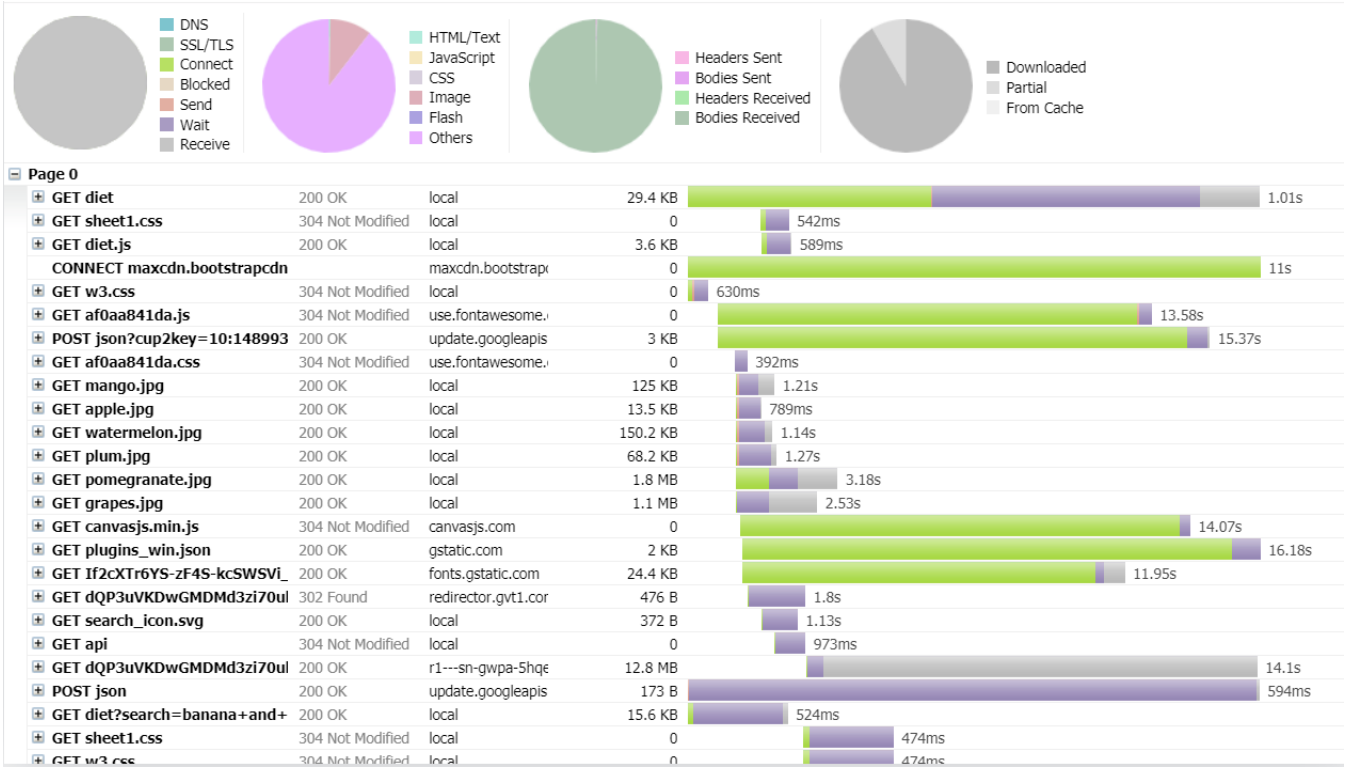




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54577004/fafa018c

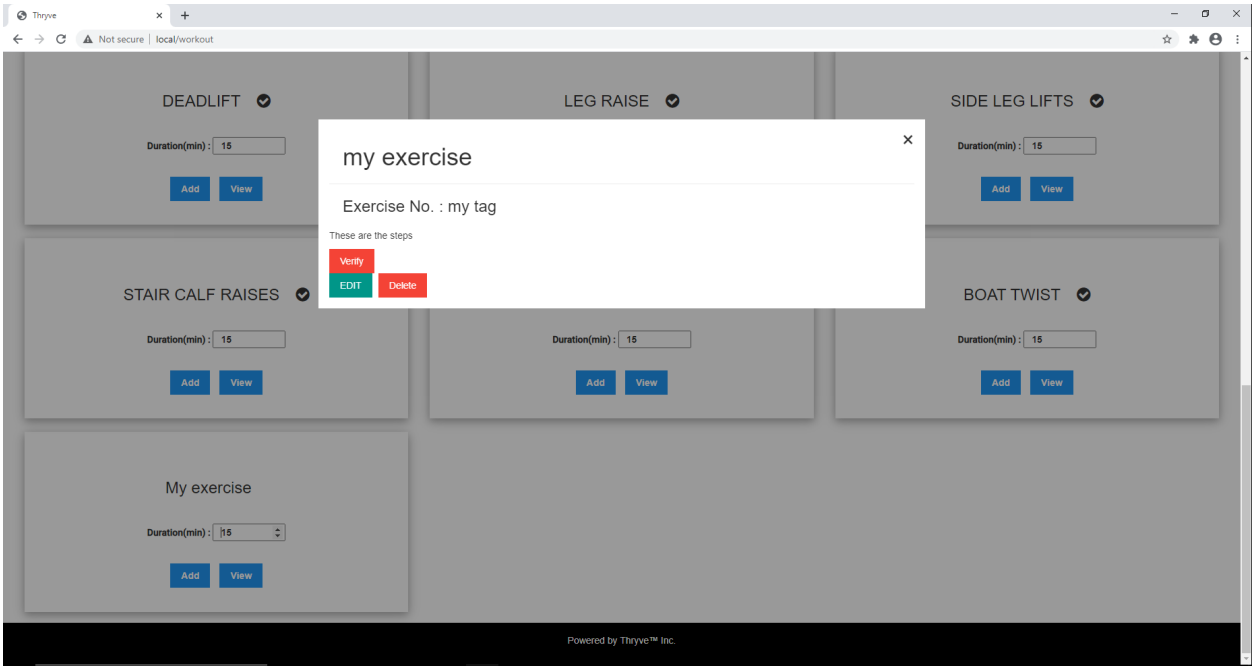
2.3.3.11 Verify Custom Dish of User by Admin

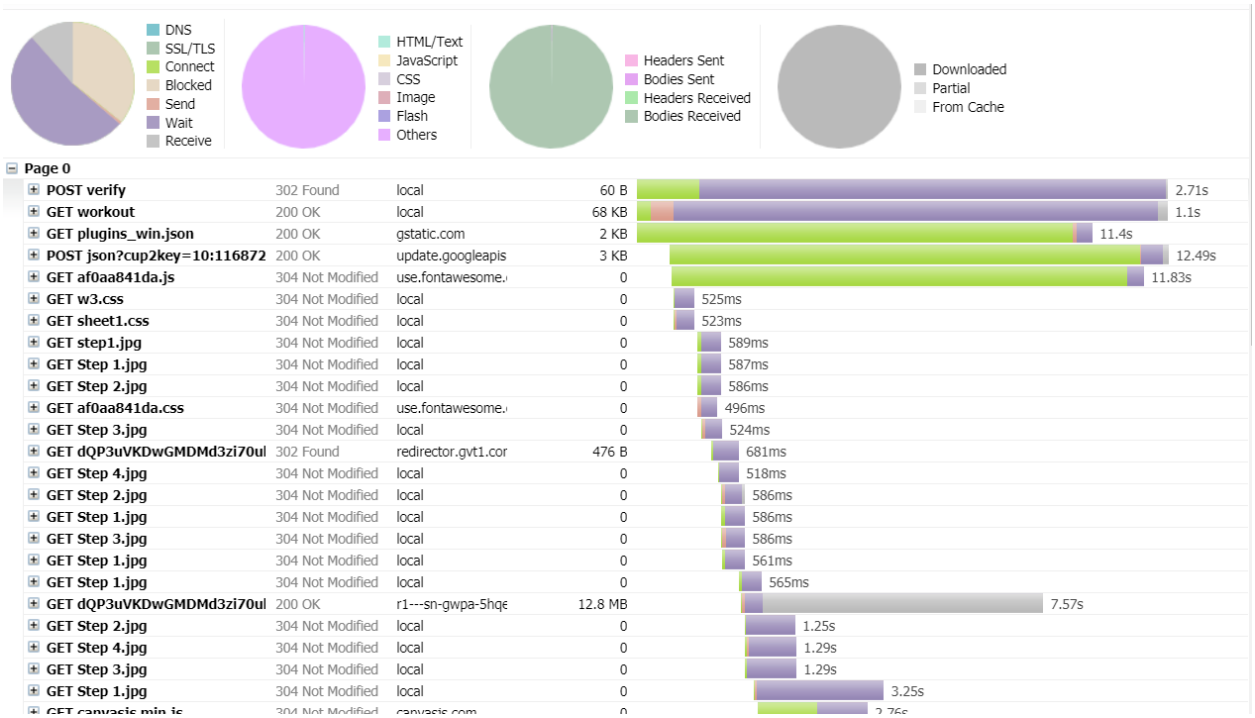




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54577183/ff6c1015

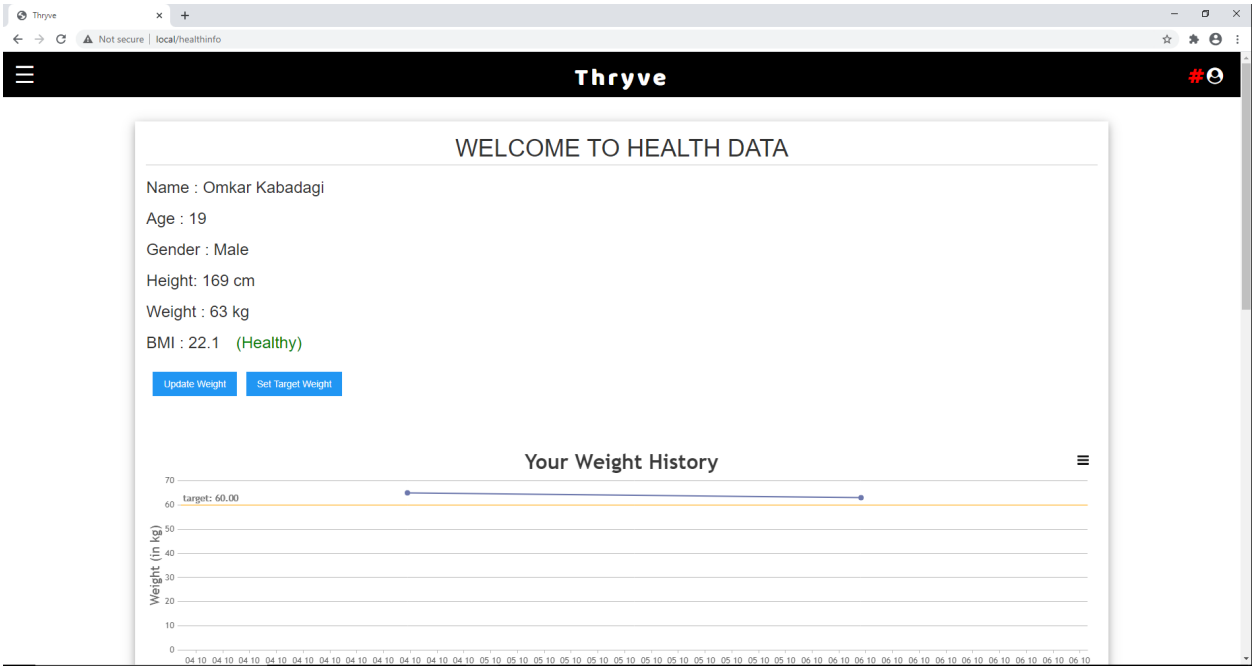
2.3.3.12 Verify Custom Exercise of User by Admin

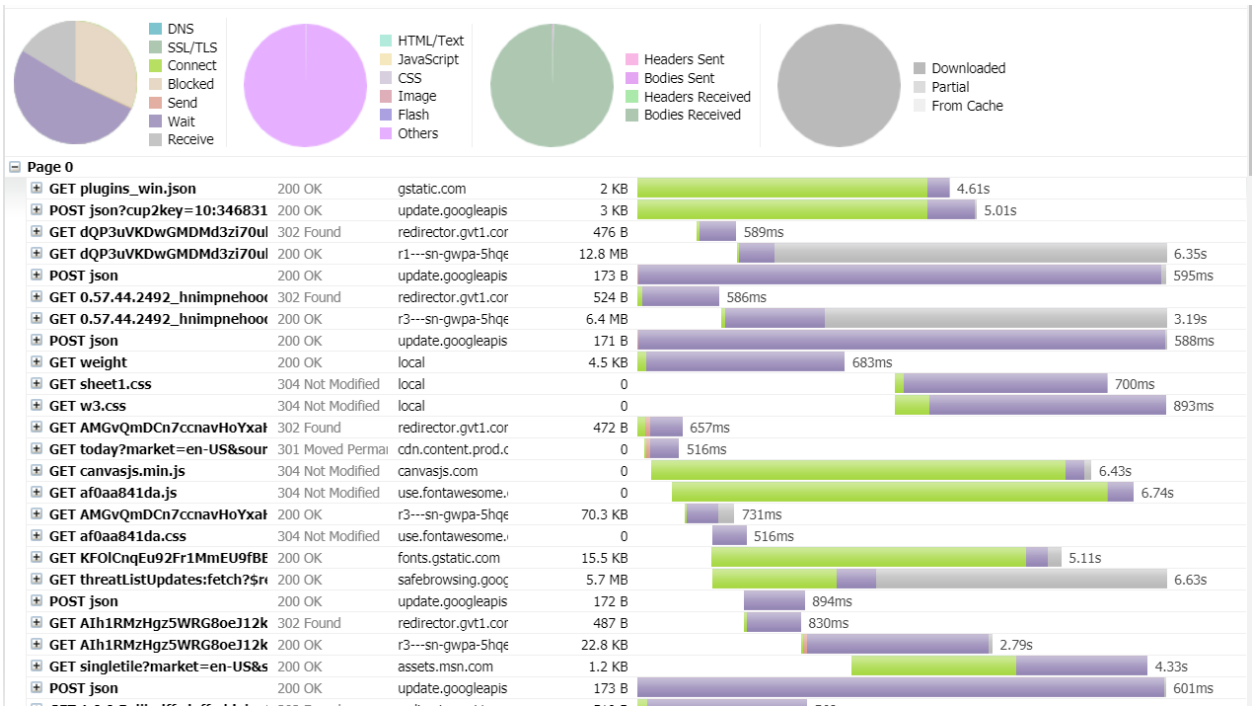




Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54578280/9a40f28a

2.3.3.13 Health Info, Update Weight, BP, Sugar





Link to test results:
https://app.crossbrowsertesting.com/public/i7a72cdef679cf65/livetests/54577393/590c25c7