CSCE 5430 SOFTWARE ENGINEERING

11/16/2020

Group Name: Mystic Gang

Group Members: Abhinav mamidipelly

Harshavardhan reddy goli

prakyath reddy kandimalla

udhaya kumar gutta

Image Based Food size and Calorie Estimation

Using CNN Classifier

**Contents**

[1. REQUIREMENTS 7](#_Toc56456623)

[1.1. Functional requirements: 7](#_Toc56456624)

[1.2 Non-Functional requirements 9](#_Toc56456625)

[2. UML DESIGN 10](#_Toc56456626)

[2.1 Class diagram 10](#_Toc56456627)

[3. SEQUENCE DIAGRAM 11](#_Toc56456628)

[Sequence diagram for User Login 11](#_Toc56456629)

[Sequence Diagram for Admin Login 12](#_Toc56456630)

[Sequence diagram for User Registration 13](#_Toc56456631)

[13](#_Toc56456632)

[Sequence diagram of System 14](#_Toc56456633)

[4. USE CASE DIAGRAM 15](#_Toc56456634)

[5. TEST CASES 16](#_Toc56456635)

[6. USER LOGIN TEST CASE FORM VALIDATIONS: 22](#_Toc56456636)

[6.1 User Login Action: 23](#_Toc56456637)

[6.3 ADMIN LOGIN ACTION: 24](#_Toc56456638)

[6.4 TRAINING THE DATASET 25](#_Toc56456639)

[7. WHO HAS CODED WHAT COMPONENTS? 26](#_Toc56456640)

[8. USER MANUAL AND INSTALLATION 27](#_Toc56456641)

[8.1 Admin Login Success case: 31](#_Toc56456642)

[8.2 Admin Failure cases: 34](#_Toc56456643)

[8.3 User Registration Success case: 36](#_Toc56456644)

[8.4 User Registration Failure cases: 38](#_Toc56456648)

[8.5 User Login Success case: 41](#_Toc56456649)

[8.6 User Login Failure case: 44](#_Toc56456651)

[9. Limitations faced during development and Feedback received during the code inspection session and actions taken. 46](#_Toc56456652)

**Members contribution Table**

|  |  |  |
| --- | --- | --- |
| **Member name** | **Contribution description** | **Overall Contribution (%)** |
| Abhinav Mamidipelly | Use Case diagram, Scope, Overall Flow of Document, updated diagrams based on feedback | 25 |
| Harsha Vardhan Reddy Goli | Test Cases, User Manual, Overall Flow of Document,  New testcases, Test data errors | 25 |
| Prakyath Reddy Kandimalla | Functional and Non-Functional Requirements, Installation | 25 |
| Udhaya Kumar Gutta | Feedback and Document Review | 25 |

# **REQUIREMENTS**

## Functional requirements:

**ADMIN**: Main user of application

**USER:** An end-user of the application will get to know the calories of food.

1. **Admin Login**:

Login page of Admin, to this page admin, can navigate by **Homepage > Admin** option. For Admin, there is no need for the register; the user id and password are fixed in the database at a time manually. After entering the user id and password admin need to click on the ‘Login’button. The next action method will connect to the database and verifies user id and password; if given login data is valid, then Admin will get the admin home page window.

1. **Upload Calorie Dataset**:

Our application calculates food calories based on food identification from images; the application must maintain food calories from food items. For this, the admin will upload an excel file which has food name and calories per gram. After logging into the application, the admin will upload the dataset, and this data gets stored in the database.

1. **View Dataset**:

After uploading a dataset of food calorie, the admin can view the dataset for verification. Admin after login, admin can click on the view dataset option, then the application will retrieve data from the database and present in tabular format.

1. **User Registration**:

To use users' functionalities, one should create an account and log into the user portal. To create an account, a user should register by providing personal details like Name, email, contact, and password. These details will store in the database, and after registration, the user will get an alert box of 'Successfully Register.

1. **User Login**:

Users can navigate to the user portal after authenticating by login option. Users can enter email id and password, and then the system will verify whether the account is valid or not and allow the user into the portal if it’s right.

1. **Upload Image**:

For food, calorie detection, users should upload a food image and enter the number of grams. The application will return the calories of the food.

1. **Calculating calories and training dataset**:

In this feature, after uploading the image, simultaneously the user needs to give the additional information about the food. Here, it is number of grams of the food. So that our desktop application can calculate the calories of the food. We maintain the food calorie per gram in database. We will be training few food items dataset with CNN algorithm. This dataset consists of 101 food item images each food item categories consists of more than 1000 images; we’ll train the dataset with neural network and get the result.

1. **User output**:

After uploading the image, if the food image is trained in our dataset, we will get the user output in a textbox.

## 1.2 Non-Functional requirements

1. **Usability**

* Our application has a simple user interface, it has very straight forward options for both admin and users with necessary modules.
* For the users it is easy to sign up with the basic info and can easily fill up and get an account created. Even we can upload the image with a single click on the upload image button and can enter the grams of data in a textbox which helps in getting the accurate result and user satisfaction can be achieved.
* For the admins it easy to update, upload, and view the data whenever required.

1. **Performance Requirements:**

* Our application has high performance when transferring from one module to another module; the average payload time is about 600ms.
* As we train data with lots of images, calorie calculation and detection fastness depend upon the execution time, roughly back 2000ms for food detection.

1. **Accuracy:**

* As in our application, CNN is used to filter the images and train a food item with many images. If the user's image matches that of the image in the database, it retrieves the appropriate images
* We achieved an accuracy of up to 75-80% (The results are based on testing the data by uploading different images). The picture we upload should have a high resolution for better results.

# **UML DESIGN**

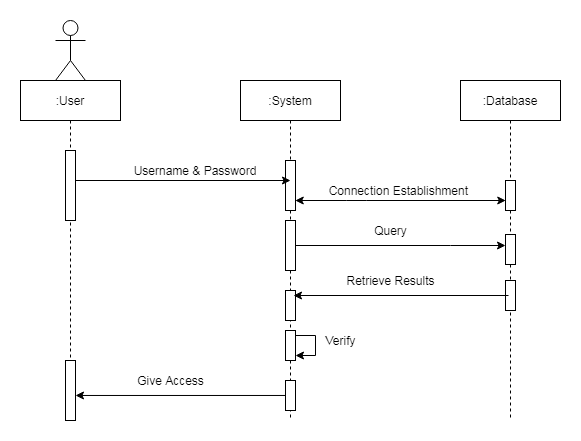
## **2.1 Class diagram**

Graphical user interface, application, Teams

Description automatically generated

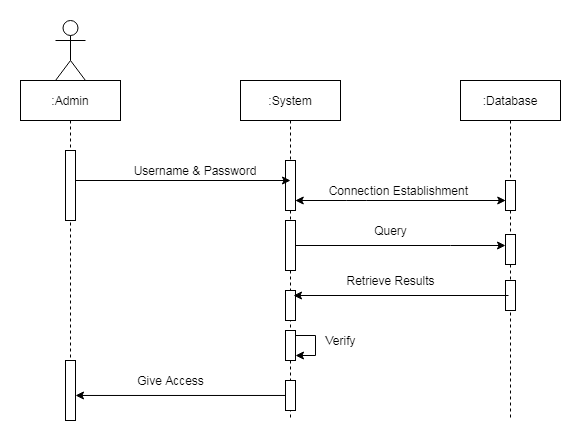
## **SEQUENCE DIAGRAM**

### Sequence diagram for User Login



### 

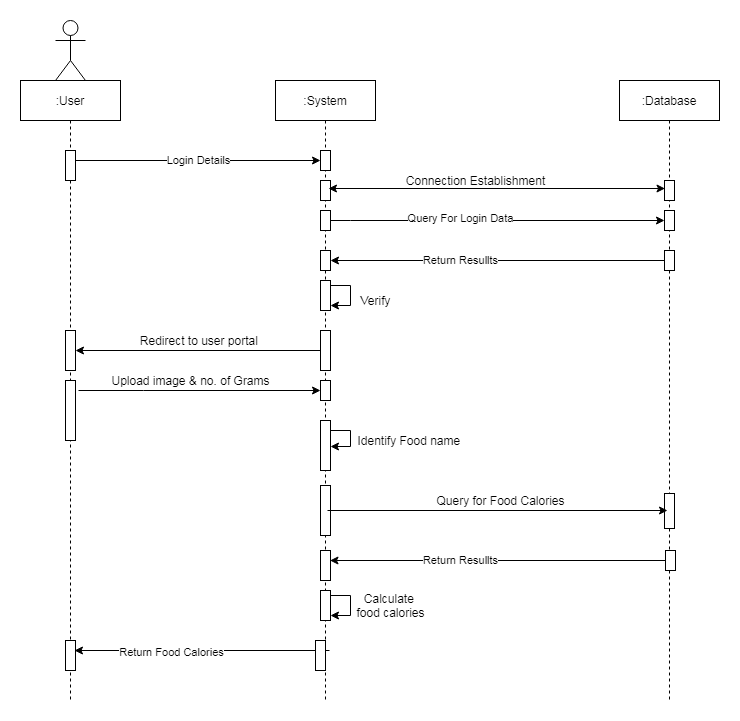
### Sequence Diagram for Admin Login



### Sequence diagram for User Registration

### 

### Sequence diagram of System



### 

# **USE CASE DIAGRAM**

Diagram

Description automatically generated

# **TEST CASES**

**Unit Testing**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Registration** | | | | **Test Priority** | | | **High** |
| **Test case description** | **Positive** | | | | **Post requisite** | | | **NA** |
| **Prerequisite** | **NA** | | | | | | | |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Result** | **Comments** | |
| Launch web Application  Runs at localhost server | | Execute manage.py by executing following command.  **‘python manage.py runserver’** | Application in localhost server | Application in localhost server | | Pass | Successful | | |
| Enter correct name, email id, password, contact no, address and hit register button | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\*\*\*\*\*\*\*  Name: Harsha  Contact no: 9403046367  Address: Frisco | Registration success | Registration success | | Pass | Successful | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Registration** | | | **Test Priority** | | **High** |
| **Test case description** | **Negative** | | | **Post requisite** | | **NA** |
| **Prerequisite** | **NA** | | |  | |  |
| **Action** | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch web Application | Execute manage.py | Application Home Page | Application Home Page | | Pass | Successful |
| Enter incorrect Email ID, and enter Name, contact no, address and password then hit register button | Name: Harsha  Email id: [harsha26gmail.com](mailto:harsha@gmail.com)  Contact no: 9403046367  Address: Frisco  Password: \*\*\*\*\*\*\*\*\*\* | Error message – Please include @ in Email ID harsha26gmail.com is missing an @ | Error message –  Please include @ in Email ID harsha26gmail.com is missing an @ | | Pass | Successful |
| Enter incorrect Contact no. and correct Name, password, Email ID, address and hit register button | Name: Harsha  Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Contact no: 9403046  Address: Frisco  Password: \*\*\*\*\*\*\*\*\*\* | Error message –Please lengthen this to 10 numbers (you are currently using 7) | Error message – Please lengthen this to 10 numbers (you are currently using 7) | | Pass | Successful |
| Don’t enter name and correct email id, contact no, address and hit register button | Name:  Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Contact no: 9403046367  Address: Frisco  Password: \*\*\*\*\*\*\*\*\* | Error message – Please fill out all fields | Error message – Please fill out all fields | | Pass | Successful |
| Enter incorrect password. and correct Name, contact no, Email ID, address and hit register button | Name: Harsha  Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Contact no: 9403046545  Address: Frisco  Password: \*\*\* | Error message –Please lengthen this to 6 numbers (you are currently using 3) | Error message –Please lengthen this to 6 numbers (you are currently using 3) | | Pass | Successful |
| Enter existing email id and its respective credentials. | Name: Abhinav  Email id: [abhinav3@gmail.com](mailto:abhinav3@gmail.com)  Contact no: 7893299141  Address: Irving  Password: \*\*\*\*\*\*\*\* | Alert message –  Email Already Registered | Alert message-  Email Already Registered | | Pass | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Login and portal** | | | | **Test Priority** | **High** | |
| **Test case description** | **Positive** | | | | **Post requisite** | **NA** | |
| **Prerequisite** | **Valid user account and credentials** | | | |  |  | |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | **Test result** | | **Test comments** |
| Launch Web Application | | Execute manage.py | Application runs in localhost server | Application runs in localhost server | Pass | | Successful |
| Enter correct email id, password, and hit login button | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\*\*\*\*\*\*\* | Login Success | Login Success | Pass | | Successful |
| Upload the image file and enter weight in grams | | Image file uploaded  Enter no. of grams:9 | Uploaded successfully | Uploaded successfully | Pass | | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **User Login and user portal** | | | | **Test Priority** | | **High** |
| **Test case description** | **Negative** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid user account and credentials** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Web Application | | Execute manage.py | Application runs in localhost server | Application runs in localhost server | | Pass | Successful |
| Enter correct email id, wrong password, hit login button | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\*\*\*\*\* | Alert:  Login Failed | Alert:  Login Failed | | Pass | Successful |
| Enter password and leave Email ID text box empty | | Email id:  Password: \*\*\*\*\*\*\*\* | Alert:  Please fill out this fields | Alert:  Please fill out this fields | | Pass | Successful |
| In the Upload food image section and enter the grams sections didn’t upload the image | | Image not uploaded | Alert:  Please select a file | Alert:  Please select a file | | Pass | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **Admin Login and portal** | | | | **Test Priority** | | **High** |
| **Test case description** | **Positive** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid Admin credentials** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Web Application | | Execute manage.py | Application runs in localhost server | Application runs in localhost server | | Pass | Successful |
| Enter correct Admin credentials | | User id: admin  Password: \*\*\*\* | Alert:  Login successful | Alert:  Login Successful | | Pass | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **Admin Login and portal** | | | | **Test Priority** | | **High** |
| **Test case description** | **Negative** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid admin credentials** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Web Application | | Execute manage.py | Application runs in localhost server | Application runs in localhost server | | Pass | Successful |
| Enter incorrect Admin credentials | | user id: admin  Password: \*\*\*\* | Alert:  Login Fail | Alert:  Login Fail | | Pass | Successful |
| Don’t enter credentials and hit login | | User ID:  Password: | Alert:  Login failed | Alert:  Please fill out the details | | Fail | Failure |
| Didn’t upload dataset in admin portal | | Upload dataset: No file uploaded | Alert:  Please select a file | Alert:  Please select a file | | Pass | Successful |
| **Test case scenario** | **Image upload** | | | | **Test Priority** | | **High** |
| **Test case description** | **Positive** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid user credentials** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Web Application | | Execute manage.py | Application runs in localhost server | Application runs in localhost server | | Pass | Successful |
| Enter user credentials | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\*\*\*\*\* | Alert:  Login successful | Alert:  Login successful | | Pass | Successful |
| Don’t upload image and give grams information | | Image not uploaded  Enter number of grams: 50 | Alert:  Please select a file | Alert:  Please select a file | | Pass | Successful |
| Upload a pizza image | | Upload dataset: .png file uploaded  Enter number of grams:12 | Output:  Food: Pizza  Calories: 87 | Output:  Food: Pizza  Calories: 87 | | Pass | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case scenario** | **Image upload** | | | | **Test Priority** | | **High** |
| **Test case description** | **Negative** | | | | **Post requisite** | | **NA** |
| **Prerequisite** | **Valid user credentials** | | | |  | |  |
| **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test result** | **Test comments** |
| Launch Web Application | | Execute manage.py | Application runs in localhost server | Application runs in localhost server | | Pass | Successful |
| Enter user credentials | | Email id: [harsha26@gmail.com](mailto:harsha26@gmail.com)  Password: \*\*\*\*\*\*\*\* | Alert:  Login successful | Alert:  Login successful | | Pass | Successful |
| Upload image and don’t give grams information | | Image not uploaded  Enter number of grams: | Alert:  NIL | Alert:  NIL | | Fail | Failure |
| Upload a pancakes image | | Upload dataset: .png file uploaded  Enter number of grams:50 | Output:  Food: pancakes  Calories: 87 | Output:  Food: beignets  Calories: 87 | | Fail | Failure |

**System testing Errors:**

**URL Mismatch Error**

When we give URL like localhost:8000/login and if it’s not matches in urls.py files, we can get this error.

Graphical user interface, text, application, email

Description automatically generated

**FieldError**

Database field mismatch from model. Given keyword 'emailid' into field. expected: age, email, gender, id, name, pwd, zip.

Graphical user interface, text, application

Description automatically generated

**TemplateDoesNotExist**

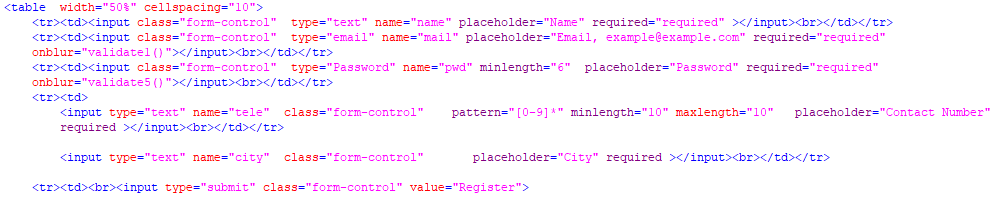
We get this error when URL redirecting and template file (html) not found, or mismatch

Graphical user interface, text, application, email

Description automatically generated

# **USER LOGIN TEST CASE FORM VALIDATIONS:**

**Password Validation testing in user account:**

****

The Above Script is written using HTML & HTML-5

* For All Text box we are checking “NULL” Validations by using the tag "required"
* For email validation we are using a tag “type="email"”
* For password validation we are using a tag type="Password"
* For Contact Number validation we are using a tag name="tele"

**User Login NULL validations**

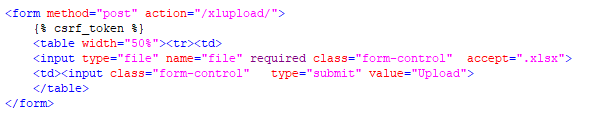
The Above Script is written using HTML & HTML-5

* For All Text box we are checking “NULL” Validations by using the tag "required"
* For email validation we are using a tag “type="email"”
* For password validation we are using a tag type="Password"

**Graphical user interface, text, application, email

Description automatically generated**

**Null Data Set Upload Validation**

****

The Above Script is written using HTML & HTML-5

* For All Text box we are checking “NULL” Validations by using the tag "required"
* For file format validation we are using a tag “accept=".xlsx"”

# **6.1 User Login Action:**

**Graphical user interface, text, application, email

Description automatically generated**

**6.2 USER SIGNUP ACTION:**

**Graphical user interface, text, application, email

Description automatically generated**

# **6.3 ADMIN LOGIN ACTION:**

Graphical user interface, text, application

Description automatically generated

# 

# **6.4 TRAINING THE DATASET**

Graphical user interface, text, application, email

Description automatically generated

The above code is to train the dataset and create a model and weight file. The key and value pairs are stored in the array and decoded\_predictions[:10] arranges the accuracy percentages in ascending order the key value is stored in the food variable after printing the maximum matched percentage the session is cleared and output is shown.

The command prompt shows how the dataset is trained (Screenshot is taken at halfway of the process as the process took 24hrs)

Text

Description automatically generated

Graphical user interface, table

Description automatically generated

After training the dataset model.h5 and weights.h5 files are created. H5 files are used to store large amount of data.

# **WHO HAS CODED WHAT COMPONENTS?**

The project has the following modules and for coding they are distributed among the team members as follows.

* User Home, User registration, User portal, get calories modules and training CNN done by Harsha Vardhan Reddy Goli
* Admin Home, Admin Login, Admin portal, Training and testing CNN modules done by Abhinav Mamidipelly
* Database connection and testing done Prakyath Reddy Kandimalla
* User Login modules and testing done by Udhaya Kumar Gutta.

# **USER MANUAL AND INSTALLATION**

The user can run code if s/he has the python 3.8 or higher version already installed or otherwise s/he can download and install, for installation s/he follow the following steps.

1. Download any version of Python 3 from the following link or the python can be downloaded from the Microsoft store.

<https://www.python.org/downloads/>

Graphical user interface, text, application

Description automatically generated

1. User need to install PostgreSQL database software.
2. User need to install Django API for web framework and server.

Text

Description automatically generated

1. The database interface used in this is pgAdmin which is PostgreSQL tool and

<https://www.pgadmin.org/download/>

Graphical user interface, text, application

Description automatically generated

1. User need to install PostgreSQL Connector API.

Text

Description automatically generated

1. User can download VScode from the below link.

<https://code.visualstudio.com/download>

1. Start Project By using following command. The terminal is opened in the VScode

**‘python manage.py runserver’**

Text

Description automatically generated

1. Open the browser through the generated link.

<http://localhost:8000/>

Text

Description automatically generated

After completion of downloading the software’s, we will execute the application in

the local machine, then the virtuoso page will be generated as follows

After downloading the required software’s we’ll be executing the application by the generated link.

This is the home page user can see at the beginning of the application.

Graphical user interface, application, website

Description automatically generated

# **8.1 Admin Login Success case**:

The admin can go to his portal by clicking Admin at the top of the application.

Graphical user interface, application

Description automatically generated

**Admin Login details:**

**User ID:** admin

**Password:** admin

After admin logged into his portal. There are three functionalities he/she can do in this they are Upload dataset and view dataset.

Graphical user interface, text, application

Description automatically generated

To upload dataset admin should navigate to **Home > Upload Dataset.**

Graphical user interface, application

Description automatically generated

After uploading the dataset Admin can also view the dataset. The data set we uploaded has 10 food items and its calorie per gram.

Graphical user interface, application

Description automatically generated

To view dataset Admin should navigate to **Home > View dataset** navigate. The food items and the respective calories per gram can be seen in the application.

A picture containing table

Description automatically generated

# **8.2 Admin Failure cases:**

Graphical user interface, application

Description automatically generated

Admin login fails if entered wrong credentials.

Graphical user interface, application

Description automatically generated

If admin doesn’t upload dataset the warning popup saying, please select a file

Graphical user interface, application

Description automatically generated

# **8.3 User Registration Success case:**

User needs to sign up with all the required credentials to get login into the user page.

# **Graphical user interface, text, application Description automatically generated**

The User Account Registration page needs to be filled out with the name, email id, password, phone number, and address.

# **Graphical user interface, text, application Description automatically generated**

After filling out the form with the correct details, we can see message Register Success, You can login at the top.

# **Graphical user interface, text, application Description automatically generated**

## **8.4 User Registration Failure cases:**

**Null validation:** If users leaves the text boxes empty a warning popup saying please fill out the fields.

Graphical user interface, text

Description automatically generated

There is Email ID validator in registration. If the email address provided was inappropriate. It does not allow the user to get registered, and also, shows a required warning.

Graphical user interface, text, application

Description automatically generated

There is password validator. Minimum length of the password needs to be provided is 6 characters or more, otherwise it will not allow the user to get registered and shows a warning message.

Graphical user interface, text, application

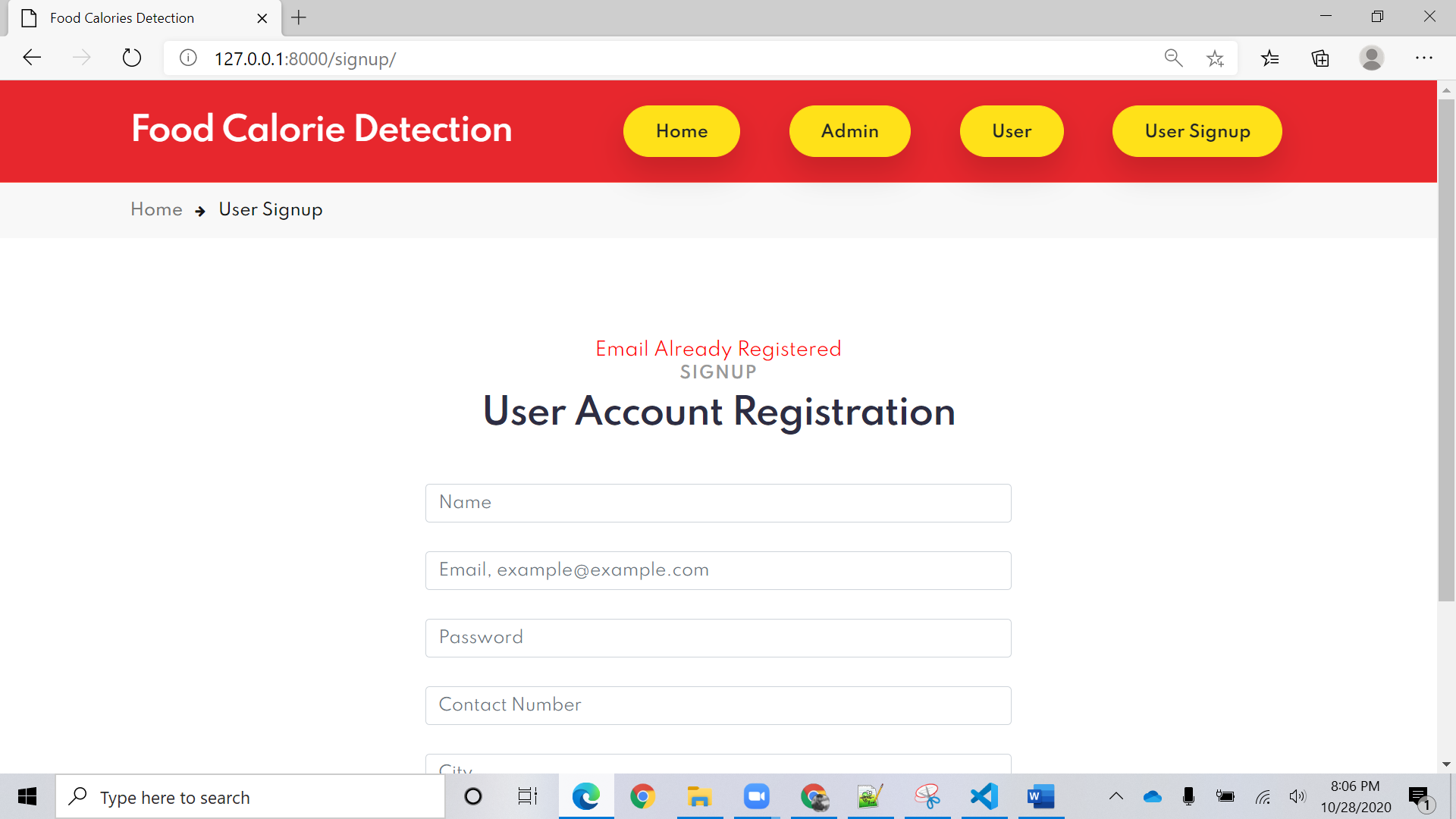
Description automatically generated

There is a Contact number validator. The number provided should be accurate that is 10 digits.

Graphical user interface, text, application

Description automatically generated

If user entered email id which is already existed an alert message pops up with the following message “Email Already Registered”.



# **8.5 User Login Success case:**

After registration user can log into user portal by his login credentials.

**Graphical user interface, application

Description automatically generated**

After user logged into his portal

Graphical user interface, text, application

Description automatically generated

# Graphical user interface, application Description automatically generated

To get calories and detect the food image we need to upload the image from the test data folder by clicking chose file it navigates to the test data folder after uploading the number of grams of food should be entered and by clicking upload button the result is shown.

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application

Description automatically generated

The result is shown as below

Graphical user interface, application

Description automatically generated

# **8.6 User Login Failure case:**

User shouldn’t leave the text box empty.

Graphical user interface, application

Description automatically generated

User should enter correct credentials to log into his portal or else login fails.Graphical user interface, application

Description automatically generated

User should upload food Image there is null validator shows warning if user doesn’t upload the file.

Graphical user interface, website

Description automatically generated

# **Limitations faced during development and Feedback received during the code inspection session and actions taken.**

The problems faced during development is we thought of using more filters and increase the accuracy and performance of the application but due to memory management we didn’t achieve what we desired for using more filters the advanced applications are used but the knowledge on the classifier is limited in the given time. During the first session we got a feedback of changing the desktop application to web application in order to handle usability testing, so we followed and started developing into web application, in next feedback sessions peer review team asked about code readability and few validation errors we rectified them in upcoming deliverables. Took feedback from the professor and made necessary changes in the document. As the development process is incremental model it has been easy foe us to change the requirements and improve the flow of development.

**References:**

The image we used in our web-application is downloaded from a following website, which is slightly edited according to our application.

<https://unsplash.com/images/food/pizza>

Images used to train data are taken from the following link

[https://www.kaggle.com/kmader/food41](https://nam04.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.kaggle.com%2Fkmader%2Ffood41&data=02%7C01%7CHarshaVardhanReddyGoli%40my.unt.edu%7Cfb936ec0c3974e9b511e08d85f03fcb5%7C70de199207c6480fa318a1afcba03983%7C0%7C0%7C637363818022377391&sdata=3p%2BQ8gg%2FQab1Kwyov551%2FDlnWYZDoeNTPRAe1xEDFZ8%3D&reserved=0)