Table of Contents

1 Project Idea

Our app meals on wheels aim to reduce food waste and climate change by delivering meals to households, homeless or individuals at home who are unable to purchase or prepare their meals. This group of people may be struggling financially or due to health conditions (i.e., physical disability) – or at health risk due to COVID.

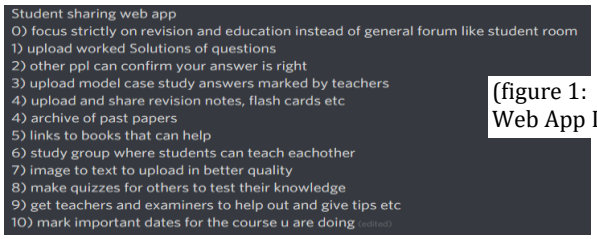
The application allows food banks/restaurants to donate ingredients and money. It has an authentication system (with email verification) where users can securely login and view their favourite meals and previous orders. The user can search nearby community kitchens and inhouse kitchen where they can order from. Upon selection, the user is asked to confirm the item and select an address to deliver it to.

We aim to achieve zero hunger (goal 2), responsible consumption and production (goal 12), and climate change (goal 13) from the UN sustainable development goals. To achieve zero hunger, we plan to reduce food waste and redistribute food that would have ended up in waste fields. Reducing food waste will also indirectly contribute to reducing climate change as fewer greenhouse gases are released into the environment. Lastly, achieving zero hunger will directly help us ensure each person is being responsible for their consumption and production.

1.1 Ideation

For the ideation phase, we have adopted the 3 important ideas of lean innovation – design thinking, lean start-up, and agile methodology. In the design phase, we organised group brainstorming sessions where each member had to presented their ideas to the group. Before the weekly meeting, each member was told to carry out research on their idea and when we all meet up again, we would discuss why we think their ideas is better. Towards the start, I did not have many ideas.

Later, I decided to focus my brainstorming topics on ideas that I would personally use or apps I would have could have had growing up such as a student platform (improving quality of education - goal 4). Upon many iterations, I decided to present the student platform (Fig. 1) to the group and it was considered as one of the possibilities.

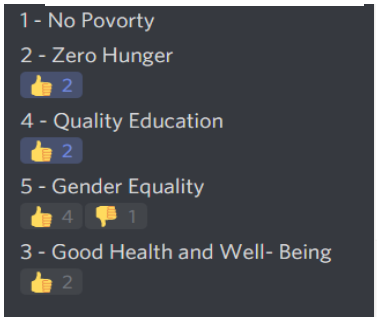


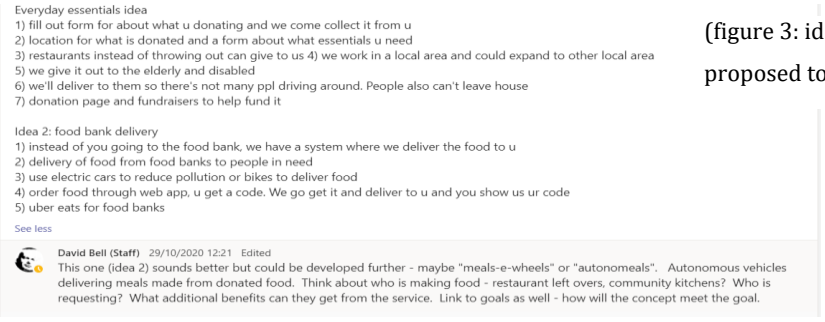
(figure 1: Student Web App Idea)

Once we had a handful of project ideas, we reviewed the list of idea and decided to carry out anonymous voting (figure 2) and gender equality was the best option at that time. The ideas were also shared with our tutor and after carefully considering the feedback (figure 3), we decided to have a change of mind and proceed with zero hunger.

Further research carried out group and I, revealed that a similar application already existed and thus, would not help us address the UN development goals. Furthermore, zero hunger covered a wider spread with a higher priority of UN goals. Zero hunger also addressed the issue that I was personally interested in and covered a variety of features to learn from.

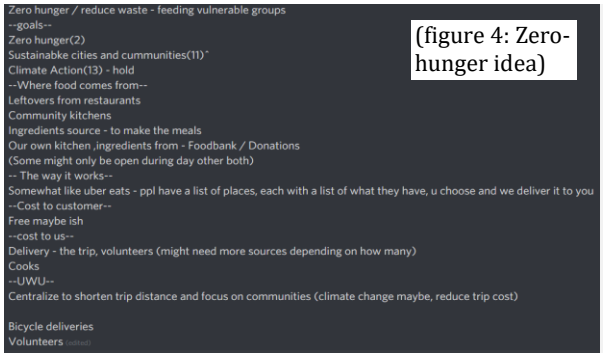
(figure 2: Anonymous voting)



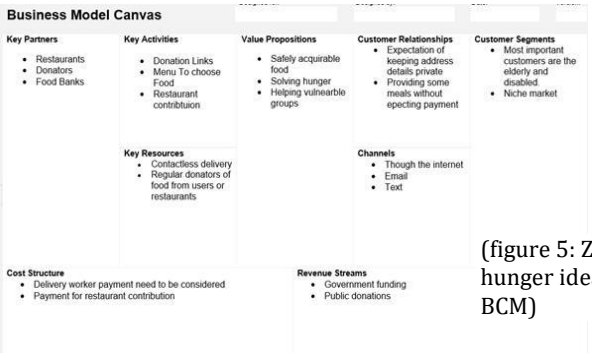


(figure 3: ideas proposed to tutor)

My idea did not get chosen as "student room" app existed which already served many of our proposed features. The zero-hunger app seemed to be much more complicated and challenging thus giving us more opportunities to explore. Later, we used the lean start-up canvas technique to create business canvas model (BCM) which tremendously helped us out in figuring key aspect from communication channels to the cost structure (see figures 4 and 5 – completed canvas).



(figure 4: Zero-hunger idea)

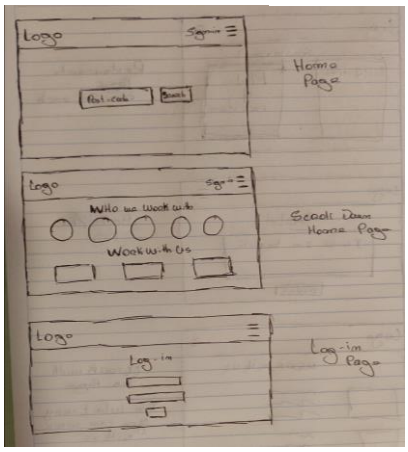


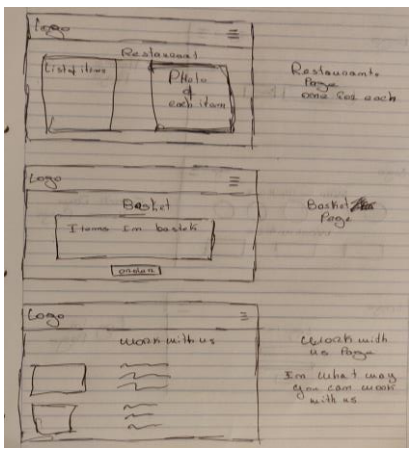
(figure 5: Zero-hunger idea BCM)

As a group, we also did the crazy 8 where each member was assigned to a team and tasked to complete the tasks. I joined with Andre and Ruthvik (see figure 6) to complete designs, as taught in usability engineering, we decided to create low fidelity prototypes (figures 7).

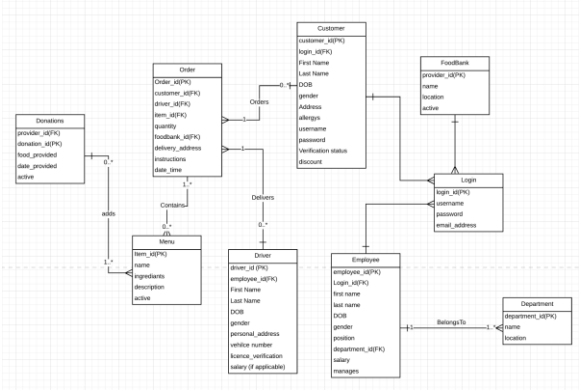
(figure 6: Crazy 8 team allocation)

(figure 7: Paper wireframes)

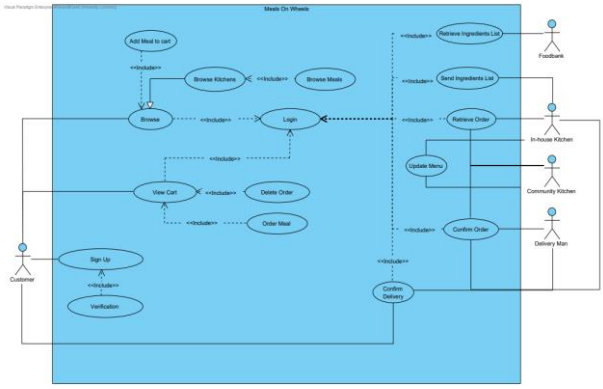




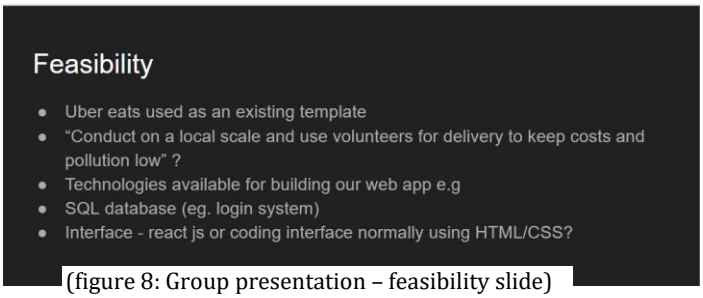
ER Diagram:



UML Diagram:



I helped out in the development of ER diagram which helped us by accelerating our understanding of our mental and conceptual designs. Lastly, the idea was pitched to the tutor in a presentation where I was tasked to complete and present the feasibility slide (see figure 8). To complete this, I researched the current applications available in the market, techniques used to implement features and what are possible improvements.



(figure 8: Group presentation – feasibility slide)

Group Research for Zero Hunger:
(*cs2001-2020_21-group10/README.md at master · BrunelCS/cs2001-2020_21-group10*, no date)
Or https://github.com/BrunelCS/cs2001-2020_21-group10/blob/master/README.md
Or (see appendix A.1)

We have planned to use agile methodology (taught in CS1004 and CS2001) as it's designed for smaller teams and gives each member an opportunity to lead and ensures we have better control over feature release, a lower risk of failure.

We could have improved our ideation process by doing multiple iterations of brainstorming sessions and by creating questionnaires (to determine public interest/demand). Furthermore, the app could have implemented a shortest route algorithm with live delivery tracking. This provided more user functionality with greater system visibility (10 usability heuristics – taught in CS2003 Usability Engineering). Similarly, the requirements could have been more refined to avoid confusion.

1.2 Requirement analysis

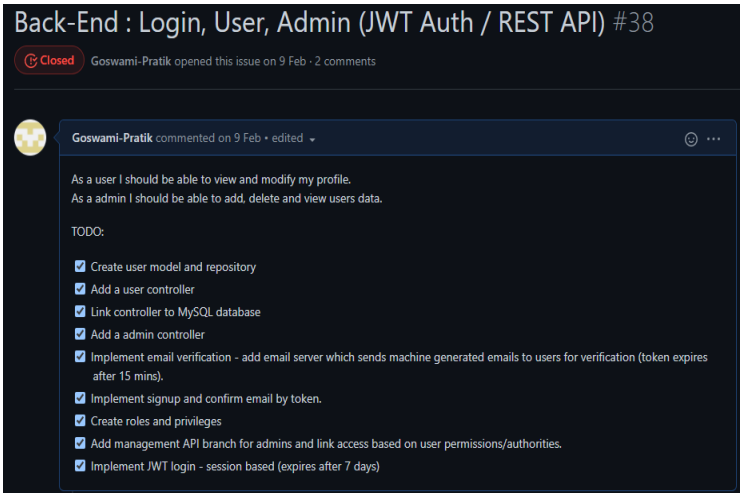
The requirements at the start, were created in a quick and dirty manner as seen in the screenshot (Fig. 9). Later on, upon feedback from our tutor, the requirements were refined into user-stories (Fig. 10), assuring us that the final product remained true to the issue we addressing. I have further broken down my stories on GitHub (Fig. 11) to allow progress tracking by scrum master.

(figure 9: Initial Requirements)

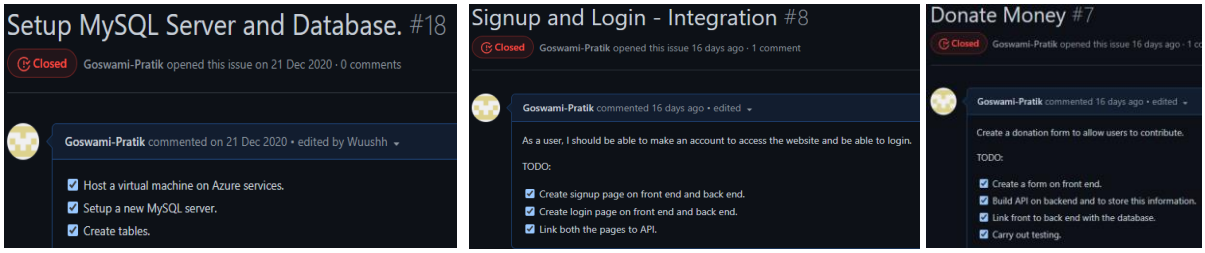
Requirements	Type	Description
Form which allows input of data	Functional	You can input food details as well as collection details
Search bar	Functional	Search for wanted products
Input product with expiry date	Functional	Allow for scanning of barcode or manual input (drop down list)
Wishlist / pending list	Functional	Put wanted items on a list and you will be alerted if avaliable
Local products	Functional	Can see products in local vicinity
Messaging	Functional	Messaging service to plan meet up / delivery
Youtube channel, Blog, google ads (All of these to advertise about our website)	Non-Functional	we need to let people know about our website.
Need a domain name, need a hosting plan	Functional	There are lot of sites which helps us to deploy our sites(Ex:- Github pages, Netlify, etc)
database of users / login system	Functional	login system to track user wish list and other things
local location of users for pickup	Functional	Local pickup of food
rough - leftover meals pickup system	Functional	pick up left overs from resturants
form for foodbank donation	Functional	donate to foodbanks

(figure 10: Refined requirements)

1. As an administrator user, I want to create an account for customers and other types of users, so that I can give access to the system.
2. As an administrator user, I want to provide separate access to the different types of users so they can use certain functions on the web app.
3. As a site visitor I want to see an introduction to what the web app is and what services are provided
4. As a site visitor, I want to be able to view FAQ, Privacy and T&C section
5. As a site visitor I want to be able to view reviews left by other customer users on the restaurants
6. As a site visitor I want to be able to return back to the home page easily
7. As a site visitor, I want to be able to make purchases through a guest account
8. As a site visitor I want to be able to view the web app on any browser (Safari, chrome etc.)
9. As a customer user I want to be shown the estimate on when my meal/ingredient will arrive to me.
10. As a customer user I want to be able to be alerted if a specific meal or ingredient becomes available
11. As a customer user, I want to log into my account with the correct details so as to be able to use the application
12. As a customer user, I want to access the different menu with food available and make a choice
13. As a customer user, I want to make a request and add myself to a waiting list of the meals I wish to receive when made available
14. As a customer user I want to be able to leave reviews about the food that I receive and the delivery process
15. As a customer user I want to be able to fill out a form that verifies if I can use the web app by verifying my income levels and other factors.
16. As a customer user I want to be able to see all of my previous orders.
17. As a customer user I want to filter the food and ingredients to fit my dietary needs.
18. As a restaurant user I want to be able to see all of the orders sent out.
19. As a restaurant user, I want to log into my account with the correct details so that I can add food that needs to be donated and deliver
20. As a restaurant user, I want to alert the driver that the food is ready for collection
21. As a restaurant user, I want to be able to view the comments/reviews left by the customers on our page
22. As a driver user I want to see a list of any food ready for collection with the address in my local vicinity.
23. As a driver user, I want to be alerted via the web app that the food is ready to be collected.
24. As a driver user, I want to alert the restaurant that I am available to pick up and deliver the respective meals
25. As a driver user, I want to alert the customer that their food has been delivered and is ready to be picked up.
26. As a driver user, I want to notify the restaurant that their meal has been delivered to the respective user
27. As a search engine, I want to access the site map so all the pages can be correctly indexed



(figure 11: Further refined – breakdown of each individual story)



(*Donate Money · Issue #7 · BrunelCS/cs2001-2020_21-group10*, no date

2 Deliverables Summary

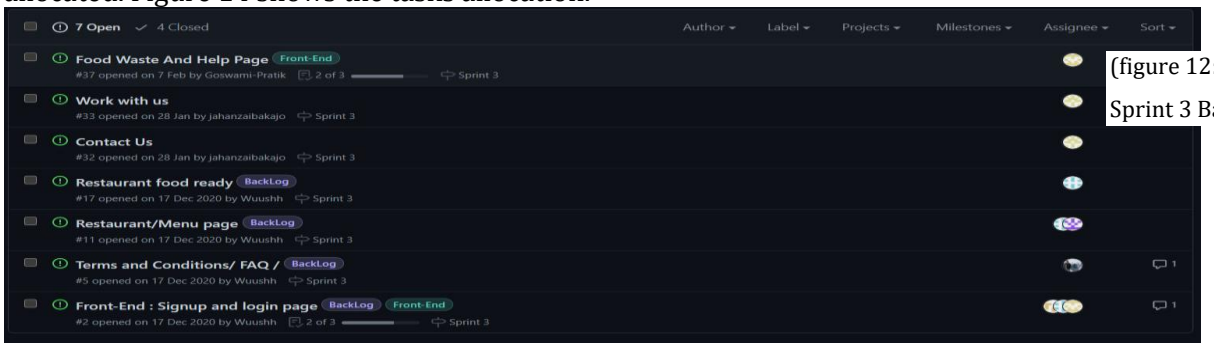
Deliverable	Done?	Lead	Individual contributions for group submissions
Placement application (CV and cover letter)	Yes	Yes	Created CV, Cover Letter and A Gap Analysis. Applied to companies – privately (Amazon, eBay, RedGate, LiquidNet, PolyAI)
Project pitch	Yes	Yes	Research and feasibility Slide (see figure 8). Assisted in presentation preparations.
Project requirements analysis	Yes	No	Contributed to the ideation process. Assisted in gathering requirement and categorizing them.
Sprint 1 Plan	Yes	Andre / Jahan	Setup Remote Virtual Machine – Ubuntu. Created page designs using Figma Assisted in the ER Diagram and Wireframes.
Sprint 1 Review and Retrospective; Sprint 2 Plan	Yes	Andre / Jahan	Setup MySQL Database with tables. Created Scripts for Table. Spring Boot Project Setup
Sprint 2 Review and Retrospective; Sprint 3 Plan	Yes	Pratik / Faizaan	GitHub Repository/Project Management (Scrum Master). Creating and Allocating Issues. **Front-End:** Started Registration and Login pages Started Food Waste Pages + Research. Modified Navigation Bar **Back-End:** Email Server + Email Verification Service. Form-Based Authentication Integrated User Rest API Implemented
Sprint 3 Review and Retrospective; Sprint 4 Plan	Yes	Ruthvik / Kieran	Linking/integrate front to back end. **Front-End:** Completed Registration and Login pages + integration and validation. Completed Food Waste Pages – Problem/Facts and How to Contribute Donate page: Money form (with validations) + integration Custom Navigation Bar. Error and Validation Added **Back-End:** JSON Web Token (JWT) Authentication – session based. Spring Security Integrated – Login and Registration. Custom Roles/Privilege Developed – USER / ADMIN. Donation form REST API implemented.
Group presentations	Yes	N/A	Delivered a presentation to the tutor.
Application demo to tutor	Yes	Team 10	Full app integration and got application ready for demo.

3 Project Management

Towards the start of the year, we split the group work accordingly where each member of the group had at least one task to complete. The tasks usually were assigned based on their abilities, the time required for completion and their personal preferences. Towards the start of the year, we did not function well as a group, as everyone researched their section and did not collaborate much. When we returned from Christmas break, most of the researching was completed and that is when everyone began extensively collaborating and implementing each other's code.
Towards the start of the year, the group faced difficulty adjusting to agile development in our group project. The Difficulty arose due to a lack of project management at the start hence, we began using the waterfall methodology. Later on, we received feedback from our tutor and began understanding how to create backlogs, stories. At this point - sprint 3 (early February), the group successfully transitioned to agile development.

I managed to complete all the task given to me in all the previous and following sprints however, some members of the group faced difficulty with submitting tasks on time hence, creating a backlog "spillover". This resulted in a domino effect where tasks were continuously being left over each sprint and being carried over repeatedly. The spillovers were not managed properly by previous sprint leaders and with the waterfall methodology in place of agile, resulted in stories/tasks being incorrectly distributed. The group lacked communication in the first few sprints which caused Faizaan's and my work to conflict.

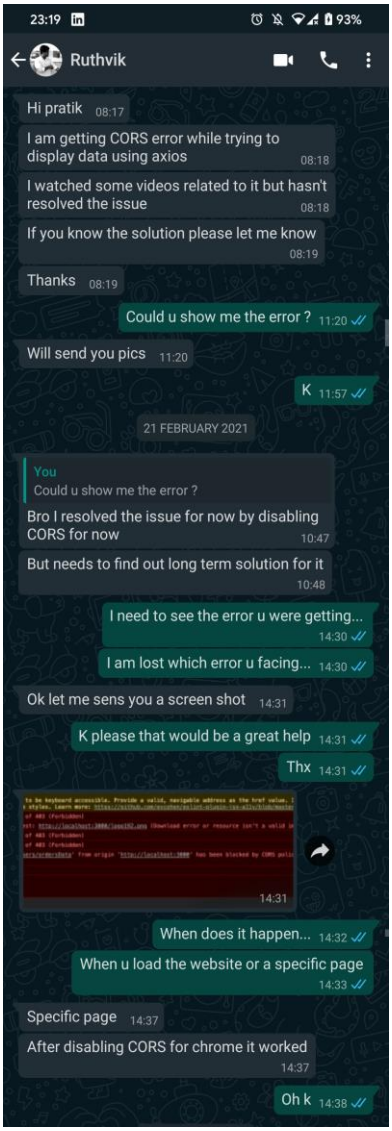
Faizaan and I lead sprint 3 where I made every effort to minimise the damage done by firstly reducing spillover from previous sprints. Tasks such as login, registration was previously assigned to a group of people, which created more stress between members and introduced more error, hence the work was reallocated to one person to reduce dependency – increasing chances of on-time completion. Later on, each member was assigned a few tasks to complete by the end of the sprint, the tasks were selected with keeping the 3-weeks sprint lifespan in mind. Each task was allocated (Fig. 12) based on a variety of factors from the individual's request, to their prior experience, to the time need to complete the task and how complex is the story. For example, Ruthvik was particularly skilled with the front-end and upon his request, we decided to allocate the profile section. I extensively provided support (Fig. 13), If any members faced any issues regarding any personal issue or challenges hindering them from completing the task allocated. Figure 14 shows the tasks allocation.

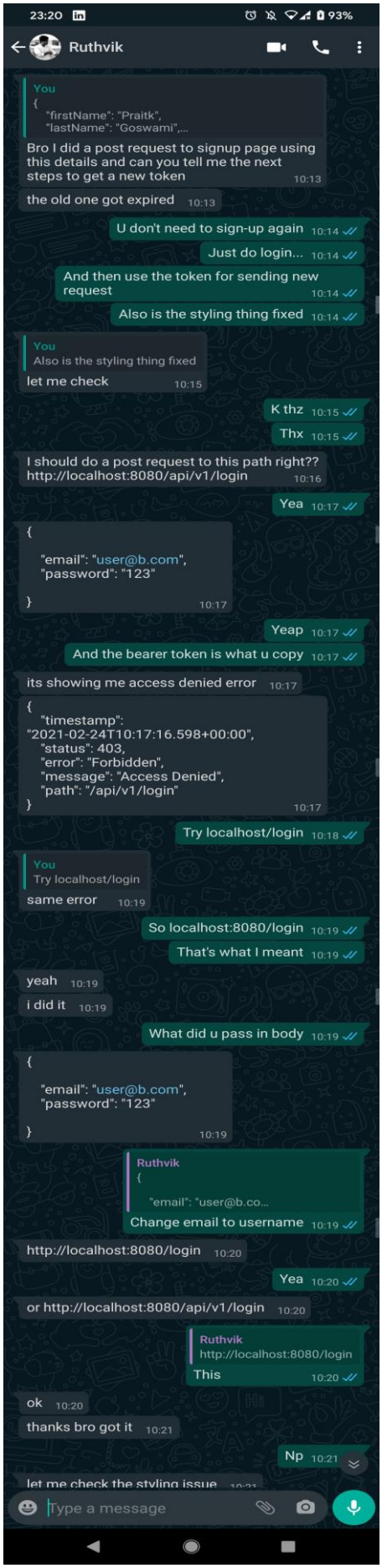


(figure 12: GitHub Sprint 3 Backlogs)

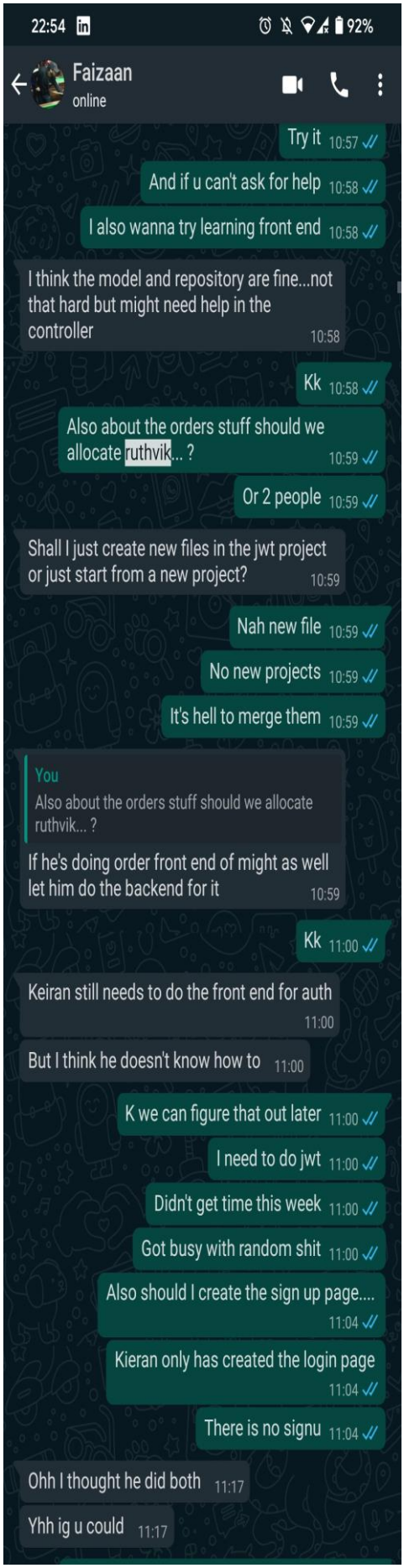
As a sprint leader, I faced challenges as few of the members were refusing to cooperate with other team members, this was overcome by being patient with them and to hear them out. I would also recommend getting help from tutors to resort this matter out before escalating it any further. Another issue I faced was members, not interaction or not showing up to meetings, to solve this I privately reached out to them and got an update. As a sprint leader, I would advise that work ethics should be strictly followed and issues should be quickly sorted out without hindering other team members.

(figure 13: Problem Solving and supporting group members)

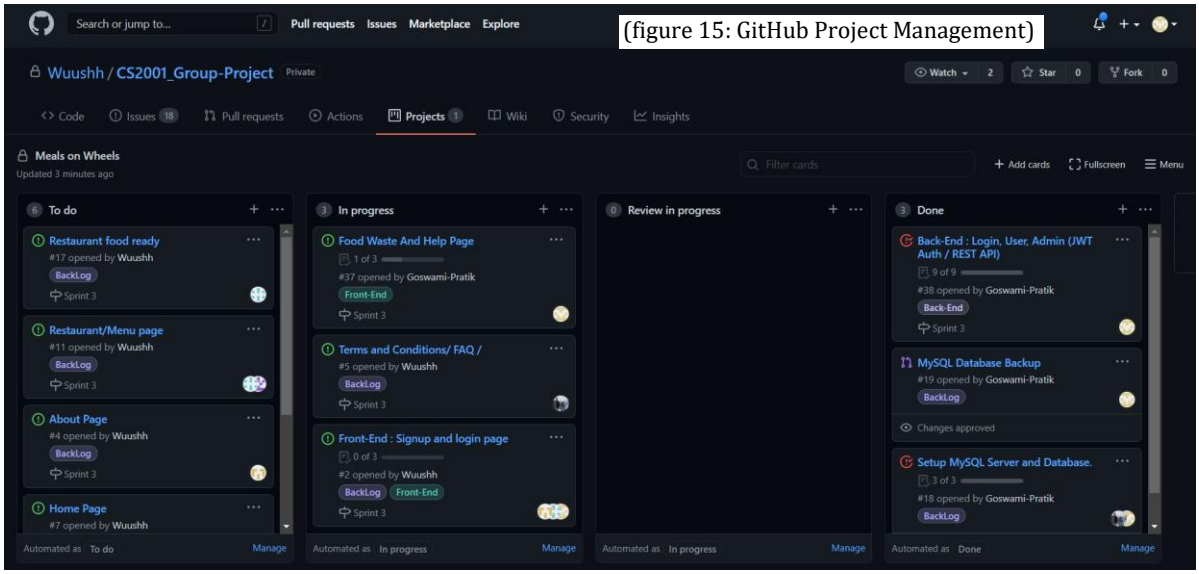




(figure 14: Task Allocation discussion)



I used GitHub project management (automated Kanban – Fig. 15) to monitor and track progress on the tasks allocated to the members. I regularly communicated with the group members and get details on how they are progressing with the task assigned (figure 13). I also ensured that important notices and new changes were discussed at the start of our weekly meetings. In future, I would have liked to have a more frequent, shorter meeting (daily standups) to see their plans and what they will be working on a day-to-day basis.



(figure 15: GitHub Project Management)

I have created a Gantt chart (Fig. 16) that highlights the crucial start and submission date for each deliverable for the CS2001 module. The Gantt chart enabled us to measure the progress of the project and provide transparency on the deliverables. It also directly ensures that the group work finished on time and is a great project management tool.

ID	Task Name	Start	Finish	Duration
1	Placement and CV	05/10/2020	09/10/2020	5d
2	Project Ideation (Labsheet 2) + BMC	12/10/2020	18/10/2020	7d
3	Group Project Pitch	16/10/2020	23/10/2020	8d
4	Sprint Planning and Estimation	19/10/2020	25/10/2020	7d
5	Git	26/10/2020	01/11/2020	7d
6	Sprint 1 Plan - Crazy 8 (UML Use-cases, User Stories and Wireframe)	05/11/2020	20/11/2020	16d
7	Sprint 1	20/11/2020	11/12/2020	22d
8	Sprint 2 Plan and Sprint 1 Review/ Retrospective	11/12/2020	22/01/2021	43d
9	Sprint 3 Plan and Sprint 2 Review/ Retrospective	22/01/2021	11/02/2021	21d
10	Sprint 4 Plan and Sprint 3 Review/ Retrospective	12/02/2021	25/03/2021	42d
11	Individual Portfolio	05/10/2020	12/03/2021	159d
12	Application Demo To Tutor	13/03/2021	25/03/2021	13d

(figure 16: Gantt Chart)

Most of the backlog's tasks for sprint 3 were completed or were near the completion stage. At the end of the sprint, I organized a session where Faizaan and I completed the review and retrospective for sprint 3. As shown in the review, most of the major features were completed and only a few tasks were left for example Integrating front-end login/signup pages to the back-end and styling the profile pages. I conversed with each member and got the progress for each of their task and what was left to implement. I later summarised and passed on the remaining backlogs to sprint 4 scrum masters and answered any of their questions. I believe that sprint 3 has been one of the most productive sprints

Sprint 3 Review/Retrospective:

Task	% completed	Notes
Food Waste Pages - (Problem/Facts, Contribute)	100%	N/A
Help Page	50%	Front-End: Added money donation form + link it to back-end Back-End: Create tables and controllers for both money and food donation.
Login / Signup Page (Front End)	66%	Link front end to backend using JWT.
Terms, Condition and FAQ	100%	N/A
Back End : Login / Registration (JWT + Permission based)	100%	N/A
About Page	100%	N/A
Home Page	90%	Completed but search bar needs backend
Find Food Page - (individual restaurant pages)	40%	Need to style the page and link with the back end.
Profile Page - Check Profile and Orders	80%	Need to adjust styling and link frontend with back-end to retrieve data from the database.
Contact Us	25%	Needs to create enquiry form and link with the backend. Generate automated email - back end.
Work With Us	0%	Start on it next sprint.

In sprint 3 review and retrospective I have also made the group complete a self-assessment where we talked about what went well, what should we start doing and what should we stop doing for the future sprints. This allowed us to self-reflect as we agreed to stop meeting impulsively and to stop creating many versions of the same file. We also discovered that we should continue assisting others, doing more testing and continue organizing GitHub projects.

Group Self-Reflection:

Start - what should we start doing ?:

- Create more frequent demos to ensure the team is on track with deliverables.
- Uploading other non technical files as a proof of our work.
- Meeting frequently(allocate new sessions if needed) for team meetings/work.
- Each meeting starts with each member discussing their tasks.

Stop - what should we stop doing ?:

- Stop creating many versions of the same file/folder.
- Stop meeting impulsively.

Continue - what should we continue doing ?:

- Helping others to distribute workload between team members.
- Assisting others with issues they are facing.
- Organizing on github and communicating via whatsapp.
- Doing testing and designing for front end pages.

Sprint 3 Review / Retrospective :

We have transitioned from waterfall methodology to an agile methodology this sprint. The group has gotten organised with work assignments, stories and workflows, as more team members have been made aware of their responsibilities. A greater number of members have started taking interest and been actively engaging compared to the last sprint. More regular commits are also taking place.

In conclusion, sprint 3 has been a vital point for the group project as at this point where everything began to come together. In future, I would have like to have more frequent, shorter daily stand-up, more organisation on GitHub from the beginning and more frequent product demo to keep the group on track with deliverables. Furthermore, there is always room for improvement in groups ethics and working professionally. As a sprint leader, I would recommend that conflicts should be resorted by being patient, hearing them out and by reaching out for assistance from other members of the group or your mentors/tutors. Also, regular group reflection tremendously assists me with sorting out areas of improvement and how better to support my team.

4 Software Design, Development and Testing

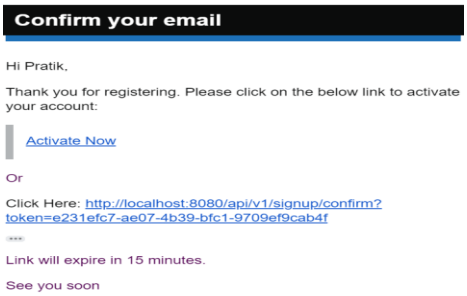
User Story: (Functional) As a site visitor, I want to be able to register a new account.

Registration Page Blackbox Testing:

(1) **Description:** to test if the user can a standard account.
Payload:

```
{
    "firstName": "Praitk",
    "lastName": "Goswami",
    "dateOfBirth": "2001-06-02",
    "email": "goswp004.304@gmail.com",
    "password": "123123",
    "address": "Malvern Avenue, Harrow, London, UK",
    "phoneNumber": "07543543435",
    "allergy": “None”
}
```

Expected: a new user is added to the database and the user redirected to the login page.
Actual: the user is redirected to the login page and the account is successfully added. Email verification is sent to the user.

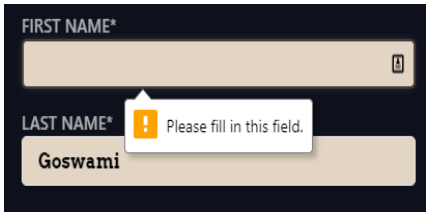


Status: PASS

(2) **Description:** to test if the user can create an account with incorrect input/credentials.
Payload:

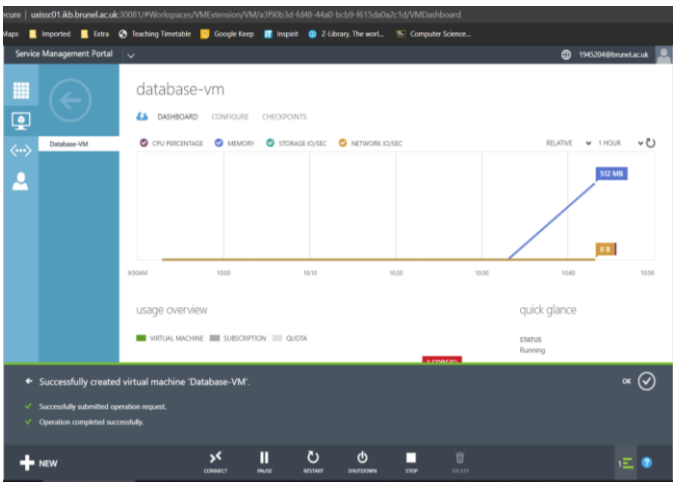
```
{
    "firstName": null,
    "lastName": "Goswami",
    "dateOfBirth": "2001-06-02",
    "email": null,
    "password": "123123",
    "address": "Malvern Avenue, Harrow, London, UK",
    "phoneNumber": "07543543435",
    "allergy": “None”
}
```

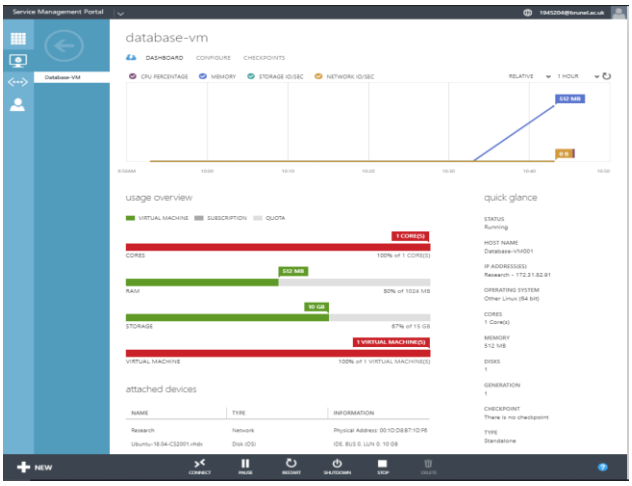
Expected: should return error message “field is required” error.
Actual:



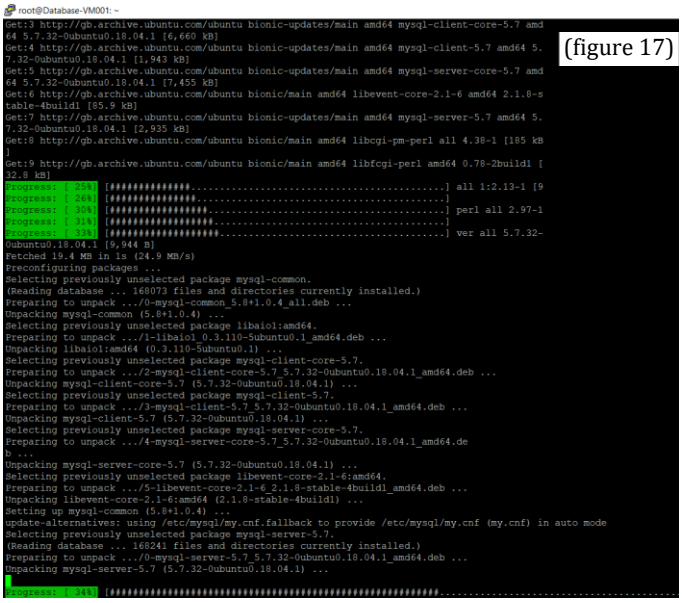
Status: PASS

I had to first request permission from the admin department. Upon approval, I followed the guide provided in module CS2001 and got remote access into the server using a tool called putty.

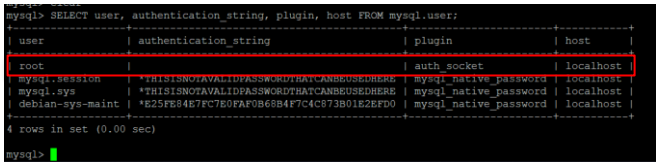




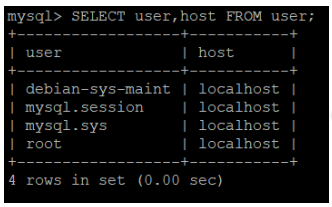
Once I connect to the server, I found it challenging to use the console interface compared to the ordinary windows user interface. In the console, I installed the necessary packages and checked networking status (using CS2005 – Networking lecture – Fig. 17). Once MySQL was installed, I created a new user (see Fig. 18). After the database was configuration, I used workbench to create a new user table with the necessary fields. This is where user data will be stored.

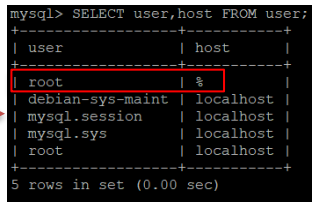


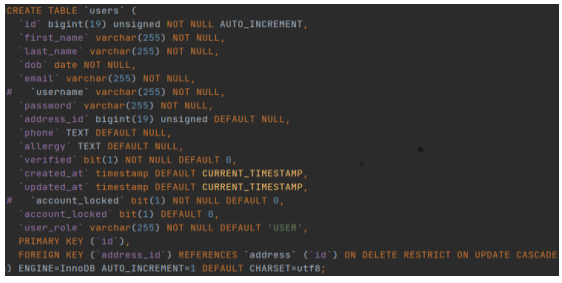
(figure 17)



(figure 18)

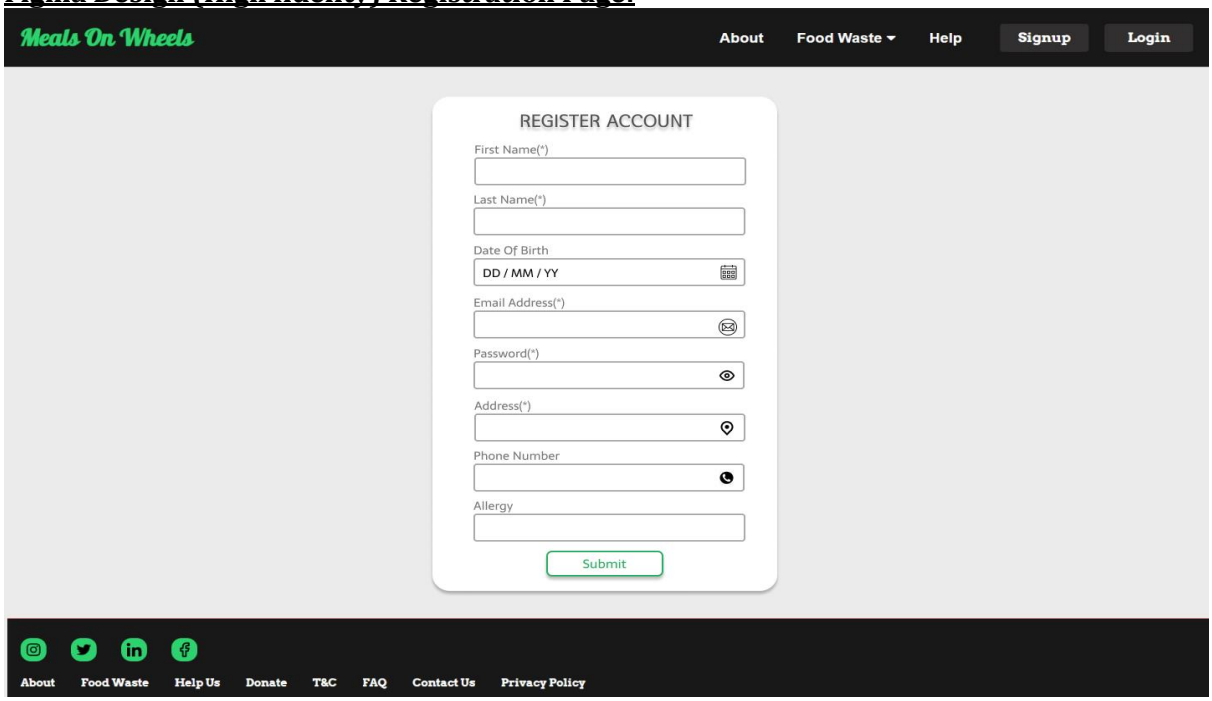




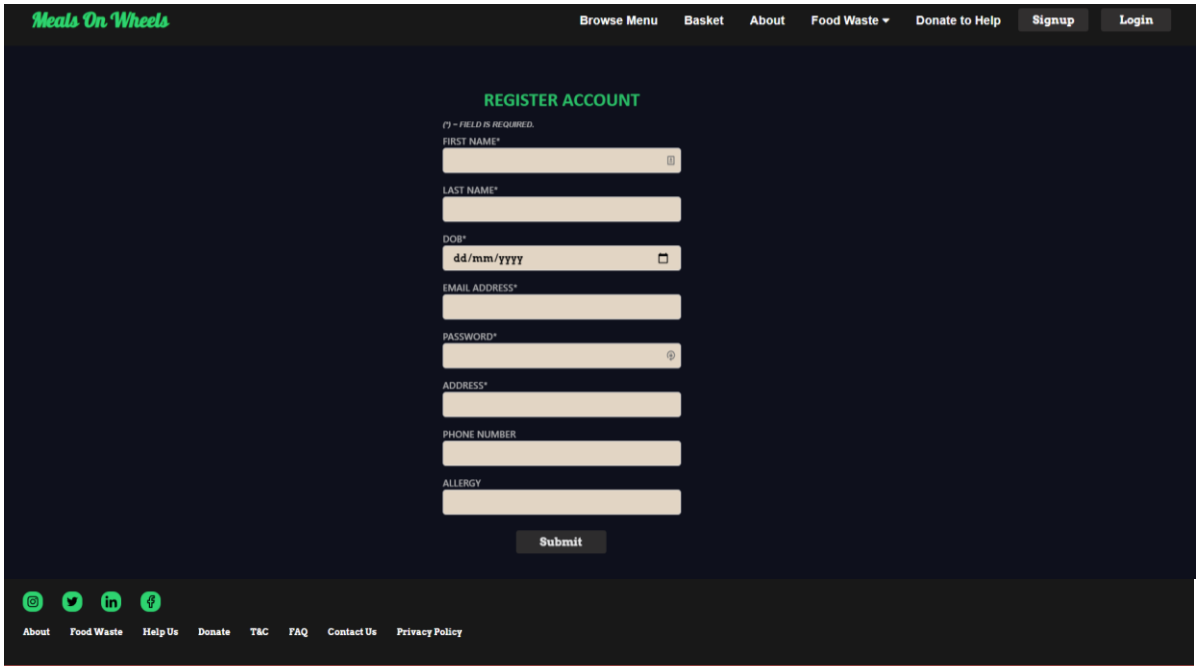


To develop the frontend, I firstly used Figma to create low fidelity paper wireframes from crazy 8(see figure 7). I later used those drawings to build the frontend using react, bootstrap, HTML, CSS and JavaScript. We decided to use react and bootstrap for frontend development as they provided pre-made components and libraries, hence it's quicker and much easier to implement.

Figma Design (High fidelity) Registration Page:



Registration Page - Actual Implementation:



I next began working on the backend, at this point I had to follow a series of online websites and tutorials (*Spring Boot Token based Authentication with Spring Security & JWT - BezKoder*, no date; *Amigoscode | Amigoscode*, no date)

Registration Service:

```
public String signUpUsers(@Valid Users users) {
    boolean userExists = usersRepository.findByEmail(users.getEmail()).isPresent();
    if (userExists) {
        Optional<Users> currentUser = usersRepository.findByEmail(users.getEmail());
        if (currentUser.get().getAccountVerified()) {
            throw new IllegalStateException("Email already taken and verified, try Logging In.");
        } else {
            if (confirmationTokenService.checkIfValidTokenExist(currentUser.get())){
                throw new IllegalStateException("Email Taken - A valid token already exists, verify it.");
            } else {
                String newToken = UUID.randomUUID().toString();

                ConfirmationToken newConfirmationToken = new ConfirmationToken(newToken, LocalDateTime.now(),
                        LocalDateTime.now().plusMinutes(15), currentUser.get());

                confirmationTokenService.saveConfirmationToken(newConfirmationToken);

                return newToken;
            }
        }
    } else {
        String encryptedPassword = bCryptPasswordEncoder.encode(users.getPassword());

        users.setPassword(encryptedPassword);

        usersRepository.save(users);

        String token = UUID.randomUUID().toString();

        ConfirmationToken confirmationToken = new ConfirmationToken(token, LocalDateTime.now(),
                LocalDateTime.now().plusMinutes(15), users);

        confirmationTokenService.saveConfirmationToken(confirmationToken);

        return token;
    }
```

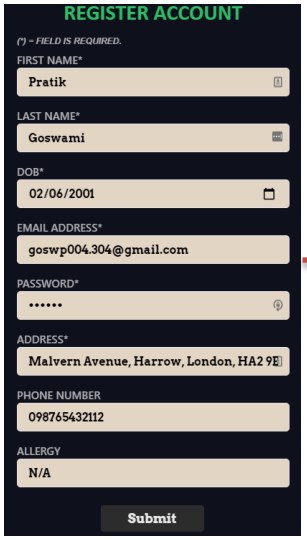
User Repository:

```
@Repository
@Transactional(readOnly = true)
public interface UsersRepository extends JpaRepository<Users, Long> {
    Optional<Users> findByEmail(String email);

    @Transactional
    @Modifying
    @Query("UPDATE Users u " +
            "SET u.accountVerified = TRUE WHERE u.email = ?1")
    int enableUserAccess(String email);

    @Transactional
    @Modifying
    @Query("UPDATE Users u " +
            "SET u.userRole = 'ADMIN' WHERE u.id = ?1")
    int giveAdminAccess(long userID);
}
```

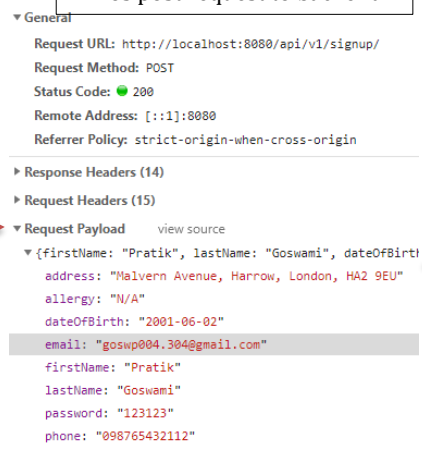
After implementing the registration controller, services, exceptions and repository, I move to integration. I have used the Axios library to send data from the registration page to the spring boot server via a post request. For registration, the database validates by checking the repository for existing accounts and decides accordingly. If the user is successfully generated, email verification is sent and the user will only be allowed to login once the email is verified. This ensures no duplicate username and allows user tracking by their email address.

USER REGISTRATION FLOWCHART

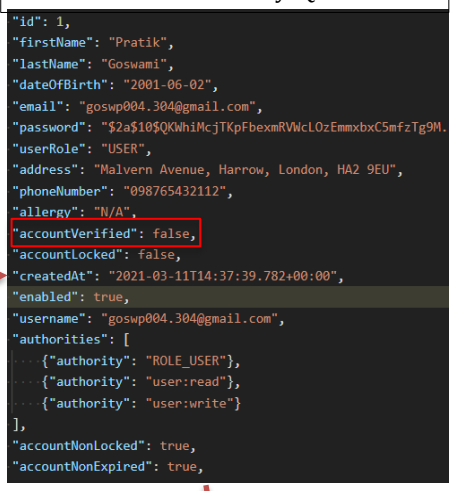
Registration Page

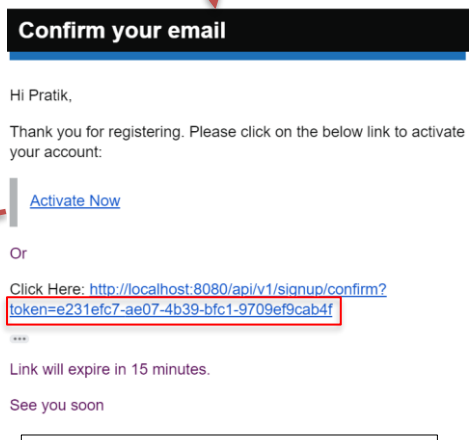


Axios post request to backend.

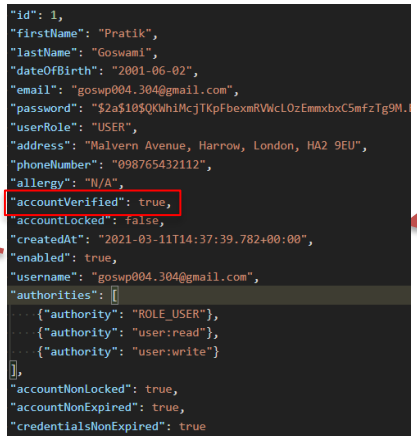


User Added to Remote MySQL Database





Email Sent to user with unique verification link.



Remote MySQL Database – account is verified.

User account create and operational.

The flowchart above portrays the system sequence when a user successfully registers a new account with us. The password is encrypted before storing into the database – performed by the registration service (see page 15). The system is implemented using spring security as it provides advance encrypting libraries and supports a wide-range of hashing algorithms.

Challenges: I had to learn how to use Azure services and how to set up the database. I further had to self-taught spring-security as I had no prior experience. Furthermore, I faced a lot of networking and porting issue with the servers. Lack of prior experience with spring boot and spring security introduced a lot of problems.

Limitations: This can be improved by fixing the cross-origin resource sharing (CORS) error. Furthermore, by implementing cross-site request forgery (CSRF) into spring securities, we ensure that one-click/session riding is prohibited. These features would improve the user's data integrity and security.

Ref: JWT: https://github.com/BrunelCS/cs2001-2020_21-group10/commit/7e79ce73c5544fd9f59eaea705c2223d93acaa80

Email Service: https://github.com/BrunelCS/cs2001-2020_21-group10/commit/52d451376038a3d028e5f2f710b63b92a0699354

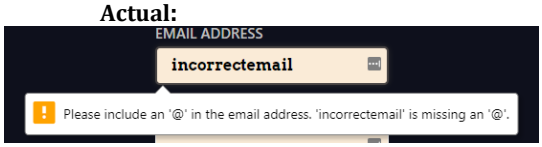
User API Spring Boot: https://github.com/BrunelCS/cs2001-2020_21-group10/commit/949dbc7d9bab21bb4b82c99aa9c390a9efbc8ad3

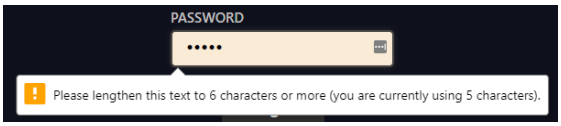
Integration: https://github.com/BrunelCS/cs2001-2020_21-group10/commit/1f59d4e7e45dc9acbc76d7e9859e58257dcb783c

User Story: (Functional) As a customer user, I want to log into my account and view my profile.

Login Page Blackbox Testing:

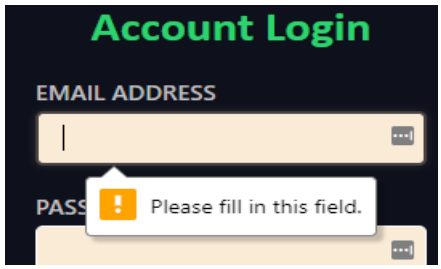
(1) **Description:** to test if the user can enter incorrect data types.
Payload: email: "incorrectemail" and password: "&^*&^"
Expected: error will show up indicating an invalid email is entered. Invalid input for the password.
Actual:





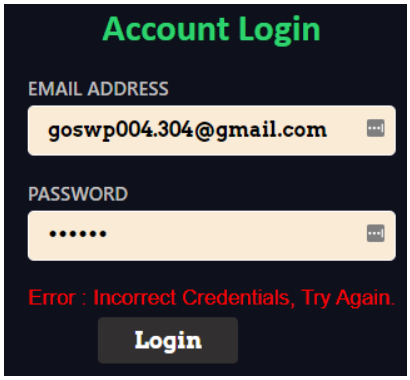
Status: PASS

(2) **Description:** to test if the user can submit without inputting email or password.
Payload: N/A
Expected: Both the field will display "Field is required" error message.
Actual:



Status: PASS

(3) **Description:** to test if the user can login without verifying email.
Payload: email: "goswp004.304@gmail.com" password: "123123"
Expected: Should not allow the user to login and display an error message.
Actual:

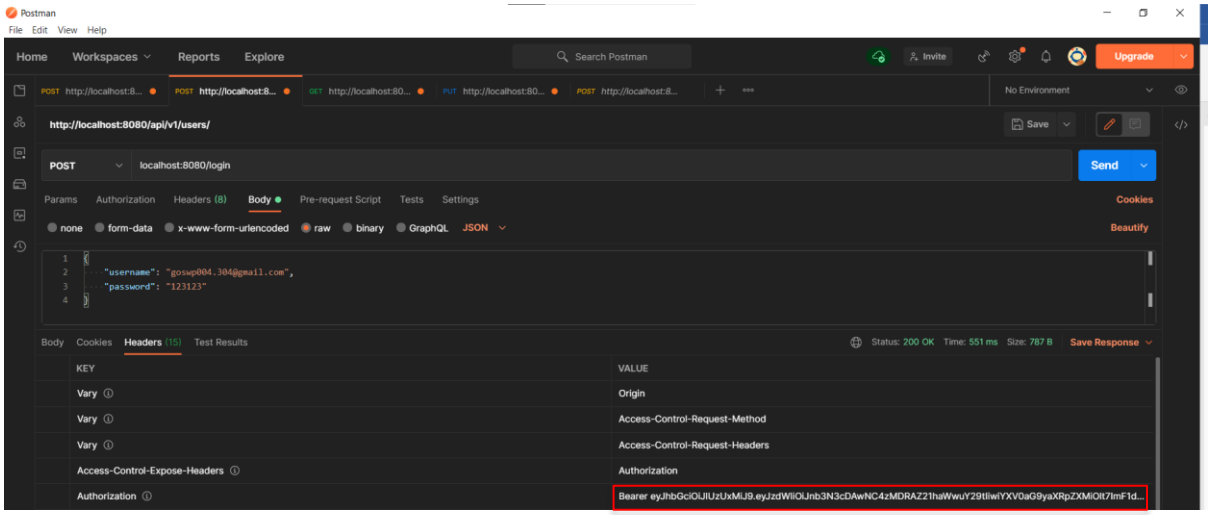


Status: PASS

(4) **Description:** to test if the user can login after verifying email.
Payload: email: "goswp004.304@gmail.com" password: "123123"
Expected: Should allow the user to login and change the navigation bar.
Actual: Allows the user to login and navigates to the homepage with a changed navbar.

Status: PASS

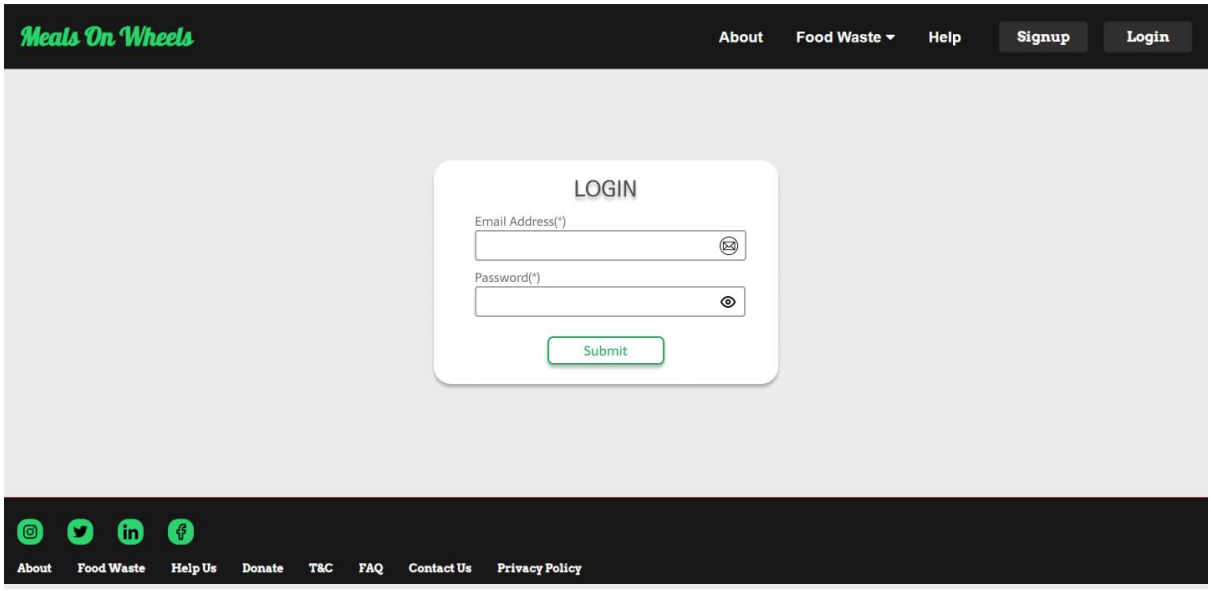
(5) **Description:** to test if the user can login after verifying email.
Payload: email: “goswp004.304@gmail.com” password: “123123”
Expected: Should return a bearer token in the response header with 200 status.
Actual:



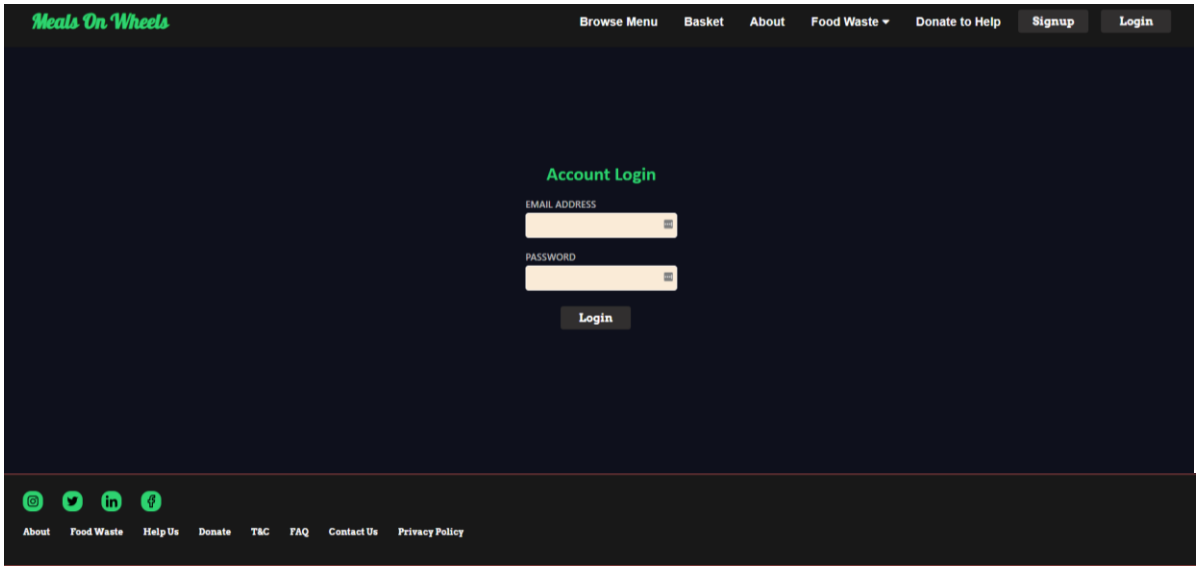
Status: PASS

I have used Figma to create designs (shown below) which assured me that I kept my pages on track with the final deliverables. The designs helped me demonstrate my end vision of the web pages and allowed other members to provide feedback.

Figma Design (High fidelity) Login Page:



Login Page - Actual Implementation:



Firstly, to develop the front-end I began by reusing code that was already implemented from my registration page. I later modified the code to add additional usestates and methods, which essentially stored the unique session token generated by the server. This token was stored in the browsers local storage and later mounted on the request header before the request is sent.

Front-End Login Service:

```
const submit = async (e) => {
    setIsLoading( value: true)
    e.preventDefault();
    const content = axios.post( url: 'http://localhost:8080/login', data: {
        username,
        password
    }).then(function (response : AxiosResponse<any> ) {
        console.log(response);
        console.log(response.headers);
        if (response.headers.authorization && response.status === 200) {
            setErrorPresent( value: false);
            setTimeout( handler: () => {
                setIsLoading( value: false);
            }, timeout: 500)
            localStorage.setItem("token", JSON.stringify(response.headers.authorization));
            localStorage.setItem("loggedIn", "true");
            return window.location = "/";
        }
    }).catch(err => {
        setErrorPresent( value: true);
        (JSON.stringify(err.response));
    })
    console.log(content);
    setTimeout( handler: () => {
        setIsLoading( value: false);
    }, timeout: 500)
```

Browser Local Storage:

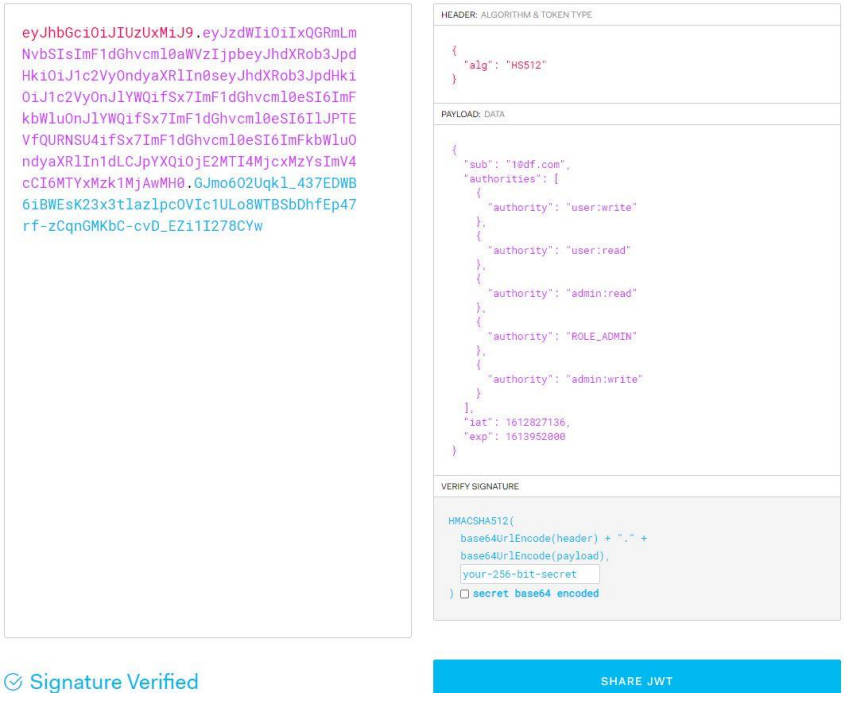
token	"Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJnb3N3cDAwNC4zMDRAZ21haWwuY29tIiwiYXV0aG9yaXRpZXMiOlt7ImF1dGhvcml0eSI6IlJPTEVfV
loggedIn	true

To develop the back-end, I firstly created new a new JWT token authenticator which used custom filters to manipulate and process the token provided. It used internal filters to obtain the current user's role and provided authorities accordingly. These authorities were later used

to enable privilege-based access to API paths and endpoints (followed online tutorials from (*Amigoscode | Amigoscode*, no date; *Spring Boot Token based Authentication with Spring Security & JWT - BezKoder*, no date)).

```java
@Autowired
public JwtUsernameAndPasswordAuthenticationFilter(AuthenticationManager authenticationManager,
                                                  JwtConfig jwtConfig, SecretKey secretKey) {
    this.authenticationManager = authenticationManager;
    this.jwtConfig = jwtConfig;
    this.secretKey = secretKey;
}

@Override
public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response)
        throws AuthenticationException {
    try {
        UsernameAndPasswordAuthenticationRequest authenticationRequest = new ObjectMapper().readValue(request.getInputStream()
                UsernameAndPasswordAuthenticationRequest.class);
        Authentication authentication = new UsernamePasswordAuthenticationToken(authenticationRequest.getUsername(),
                authenticationRequest.getPassword());
        Authentication authenticate = authenticationManager.authenticate(authentication);
        return authenticate;
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

@Override
protected void successfulAuthentication(HttpServletRequest request,
                                        HttpServletResponse response,
                                        FilterChain chain,
                                        Authentication authResult) throws IOException, ServletException {

    String token = Jwts.builder()
            .setSubject(authResult.getName())
            .claim( name: "authorities", authResult.getAuthorities())
            .setIssuedAt(new Date())
            .setExpiration(java.sql.Date.valueOf(LocalDate.now().plusDays(jwtConfig.getTokenExpirationAfterDays())))
            .signWith(secretKey)
            .compact();
    response.addHeader( name: "Access-Control-Expose-Headers",  value: "Authorization");
    response.addHeader(jwtConfig.getAuthorizationHeader(),  value: jwtConfig.getTokenPrefix() + token);
}
```

Once the user is authenticated (by class shown above), a token is generated and send back with the response header. The token contains information regarding hashing function used, the expiration time, email, the authorities present and if the signature is verified or not (see below). This token is sent to the server with each request to provided identification. A verification class (see appendix A.2) was used to verify the tokens signature and the authorities.



Challenges: I found it extremely difficult to find a full course that demonstrated how to implement JWT in spring securities with react. I faced many debugging issues (see appendix A.2) extended the expected completion date. I also had to learn many new concepts such as JWT authentication, granted authorities and internal filters.

Limitations: I could have improved this by storing the bearer token into a database to ensure the user doesn't spam the server with the login request. Furthermore, I could have implemented a password resetting system however, due to lack of time I was unable to.

Ref: Login Page: https://github.com/BrunelCS/cs2001-2020_21-group10/commit/031a20c3b219bbffdad377da688e06535d692235
Integration: https://github.com/BrunelCS/cs2001-2020_21-group10/commit/1f59d4e7e45dc9acbc76d7e9859e58257dcb783c
Login Backend: https://github.com/BrunelCS/cs2001-2020_21-group10/commit/7e79ce73c5544fd9f59eaea705c2223d93acaa80

Task: Improving usability experience for the help page.

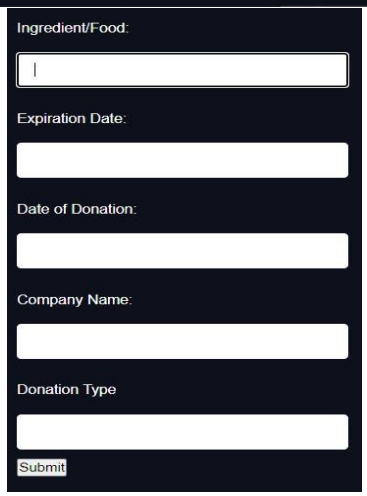
The help was poorly designed and lacked many of the crucial key validation and error exceptions. I have fixed this by modifying the code implemented by Faizaan, and have added new error handling which ensures that the required filed are filled out before submitting it. A loading animation has also been added and the submit button gets disabled while the request is being sent and processed.

I have implemented this feature keeping in mind the 10 usability heuristics (from cs2003), which aim to handle error prevention and improve the visibility of the system status. The visibility of the system is improved by adding the loading cursor and having similar fields for data input in the form allow for improved recognition rather than requiring a recall.

<u>Before Usability Improvements</u>:

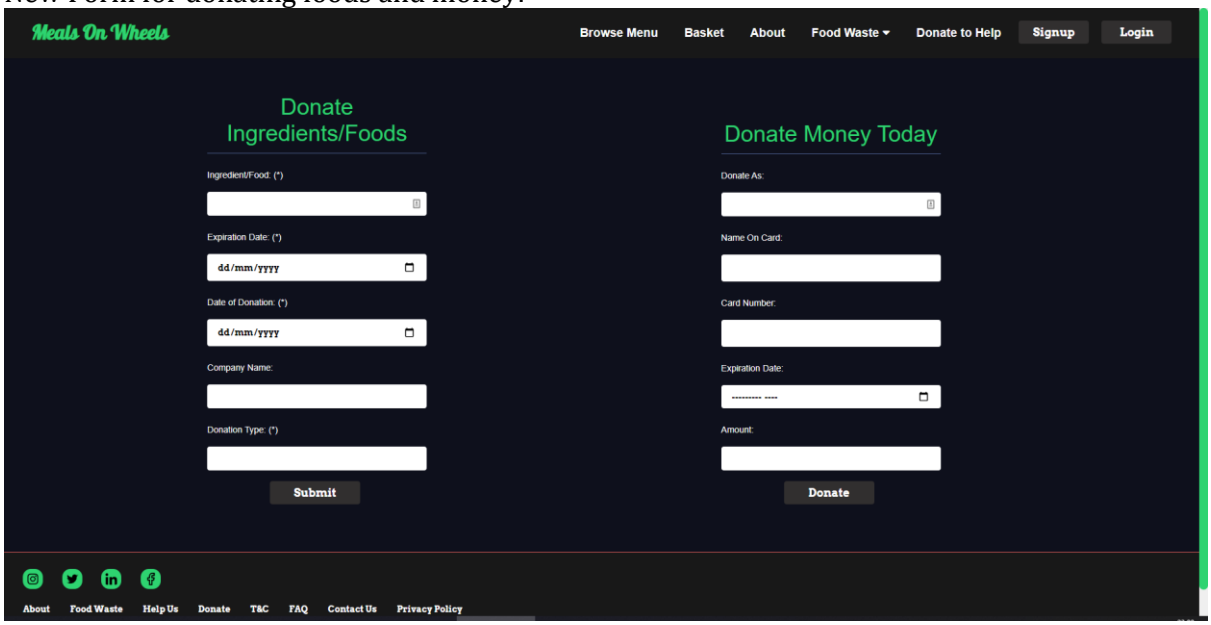
Image of the help page (styling issues, no error handling and misleading fields.)

Old Donate Food Form:

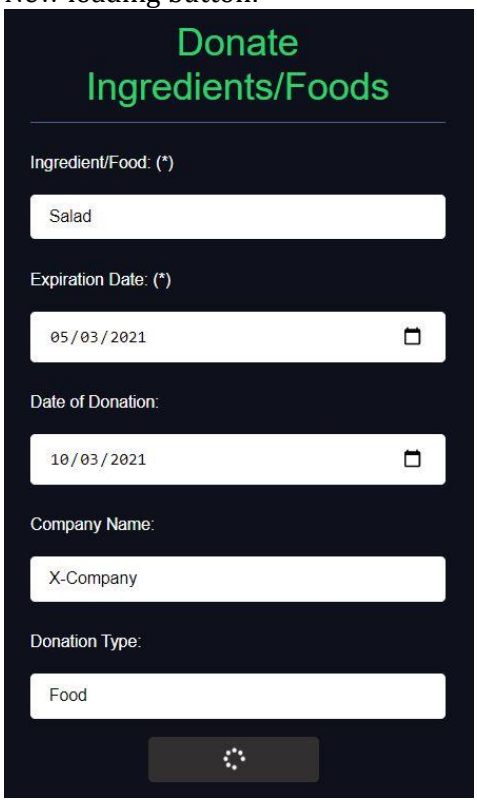


After Usability Improvements:

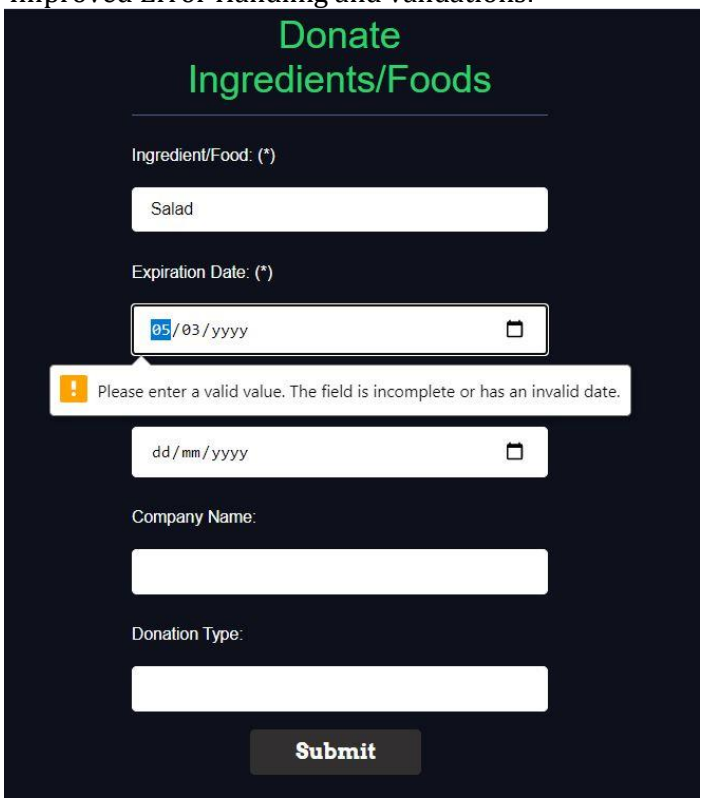
New Form for donating foods and money:



New loading button:



Improved Error Handling and validations:



In conclusion, I have managed to explore both the front and back end however, I have only managed to breakdown a handful of stories/task (see appendix A.3 for more). Throughout the project, I have demonstrated how I have implemented advance REST API and security protocols using advance tools and techniques.

5 Communications

My communication with the group was very well. I was always present for meetings and provided contributions where possible. We created new groups on WhatsApp, discord and teams to ensure that each member could backtrack old conversation and prevent miscommunications. I made contributions to project pitch, presentation and volunteered to help others out to learn something new each time. I also initiated many group conversations regarding important deliverable and issues. I have made constant effort to ensure that absent members were informed about the key discussion that took place.

I communicated any challenges that I was facing to my group and got advice on how to proceed with it. I faced many challenges when communicating with some of the group members, as they were constantly being absent which further increase the workload on the group. I have also experience unethical behaviour from a group member however, I quickly cross this hurdle by having effective communication with the tutor and got the issue sorted before it escalated.

I could have improved by being more interactive in tutor meeting from time-to-time and also by reaching out to members that we not attending the meetings. I have learnt to be more patient with my group members and have been more open to listening to ideas before dismissing them. I learnt that taking initiative actions are far better than waiting around for others to start for me. I have also improved my communication and presentation skills throughout the year. I have learnt the importance of effective communication between group members to avoid conflict later down the project.

6 Personal Development and Self-Evaluation

6.1 Personal Development

At the start of the year, I had no prior experience working with spring boot, sprint project management, react or postman for testing but after completing this project, I believe that I have improved my understanding of these topics drastically. This has directly helped me to improve my communication skills and team working skills. I have also learnt to be more patient and open to ideas before dismissing them. I believe that these skills will prove beneficial in real situations and have given me valuable insight into full-stack development. I have also improved my coding skills as previously I had little to no knowledge of REST API, spring security and spring boot thus having learnt them has vastly enhanced my current skills base.

Table Showing Personal Development by individual skills:

Skill	Before Project (1-10)	After Project (1-10)
Linux	3	7
HTML / CSS / JSX	6	9
MySQL	2	6
React	0	5
Spring boot/ Spring Security	0	8
Postman	1	8
Team Work	7	9

Project Management	6	8
Time Management	7	8
Git – version control /GitHub	3	9
Java	6	8

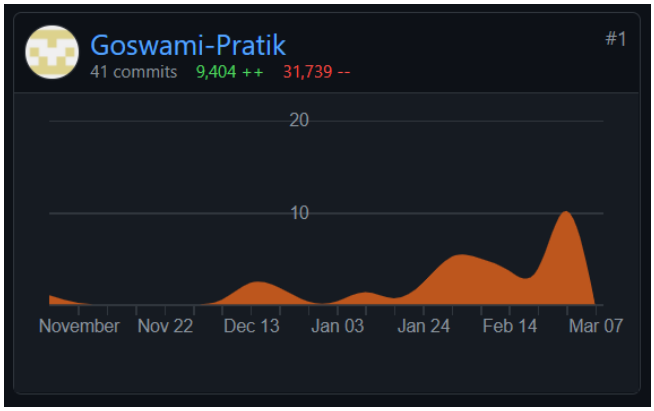
I have identified gaps in my front-end development skills in particularly the react framework. I plan to improve my skills by building project and by completing free online courses over summer breaks. I will carry out self-reflection and further improving on feedback from tutor/placement applications.

6.2 Groupwork Participation

Most of the group worked well throughout the project however, we had few members mostly being absent therefore increasing the workload on other members and had to constantly shuffle tasks between ourselves. I believe my performance towards the group was very well, as I constantly managing group members, organising meeting, ensuring that we were on track and lastly, keeping the group's morale high - providing support where possible. The table shows the scores I received for group participation. Furthermore, the commits from GitHub, indicate that I have been engaging with the group work from the start of year one till completion.

Review By	Technical skills	Attendance/Engagement	Communication	Key area to improve and strength
Jahan	10	10	10	Area to improve: Nothing that I think they need to improve on! Strength: Really helpful with great front end and back end skills
Faizan	9	9	7	Area to improve: Communcation with group (we didn't know what backend task he was doing and it clashed with mine) Strength: Good team worker and successfully integrated his tasks. Was always attending group meetings and helping out
Kieran	10	10	7	Area to improve: Dont take others tasks Strength: Knowledgable on how to solve issues and troubleshooting
Ruthvik	8.5	10	6	Area to improve: Strength: backend, helping others, even when its not about group project
Pratik	9	9	9	Area to improve: Consider others ideas more Strength: Technical Skills , Google Docs and Strong team work
Andre	9	9	9	Area to improve: Strength: He is really good in helping others with the issues others have and tries to cool things down when things get heat up
Average	**9.25**	**9.5**	**8**	

Total Commits on GitHub:



However, I believe I could improve by having more communication towards the start of the year, which could have avoided miscommunications. I could have been calmer when we got into debates as from time-to-time members of our group got into heated arguments, however, these issues were later solved.

6.3 Performance Evaluation

		LO Met (Y/N)	Summary
LO1	Plan, manage and track a substantial group project activity.	**Y**	I have managed my sprint, created plans, arranged frequent meetings to discuss their progress on the tasks assigned. I used GitHub for project management and assigning issues to group members. **Evidence (section 1, 2, 3).**
LO2	Take an open-ended problem, collect and analyse relevant information and define and refine the requirements.	**Y**	I have kept an open-mind to taking different tasks, I have contributed to requirements gathering – by analyzing their function and role and further refining them. Created sketches and designs during the ideation phase. **Evidence (Ideation and Requirements Analysis).**
LO3	Independently and systematically design, develop and test a piece of software that is data-driven and has non-trivial functionality	**Y**	I have created prototypes, carried out testing using postman, used IDEA to implement front and back-end features with project management on GitHub. **Evidence (see section 4 – software design, development and testing).**
LO4	Compare and evaluate alternative problem solutions according to the given criteria including from a technical perspective	**Y**	I have consistently iterated over my implementations and I have constantly made usability improvements changes to previously implemented pages. I have changed my solutions accordingly to fit the given criteria. **Evidence (See Sections 4, 5 and GitHub Commits).**
LO5	Effectively present, communicate and market ideas and solutions to different audiences	**Y**	I have managed to be able to deliver demos and presentation to my group and our tutor. I have communicated with my group to prepare for reports, deliverable and presentations. **Evidence (see sections 1 – project pitch and 5).**
LO6	Understand and apply the principles of professional and ethical behaviour in a group context	**Y**	I have been professional and followed ethical behaviour throughout my group work. **Evidence (see section 5 and 6).**
LO7	Reflect and learn from the group project experience.	**Y**	I have critically reflected on my skills and performance throughout the group project. The reflections could be used for future times – if I have a similar task. **Evidence (see sections 3, 5, and 6).**

GitHub Commits: https://github.com/BrunelCS/cs2001-2020_21-group10/commits/master
GitHub Project Management: https://github.com/BrunelCS/cs2001-2020_21-group10/projects/1 Or see section 3.

I believe I should be receiving an A*, as accomplished all the learning objectives set out. Furthermore, I have provided all the necessary evidence in my portfolio demonstrating my contribution to the group project. I have further shown valuable qualities from problem-solving to team-working. I believe I have managed to reflect well on my group performance (shown in peer review) and provided critical input and improvements (section 5, 6, 7).

References

Amigoscode | Amigoscode (no date). Available at: https://amigoscode.com/ (Accessed: March 1, 2021).

cs2001-2020_21-group10/README.md at master · BrunelCS/cs2001-2020_21-group10 (no date). Available at: https://github.com/BrunelCS/cs2001-2020_21-group10/blob/master/README.md (Accessed: March 1, 2021).

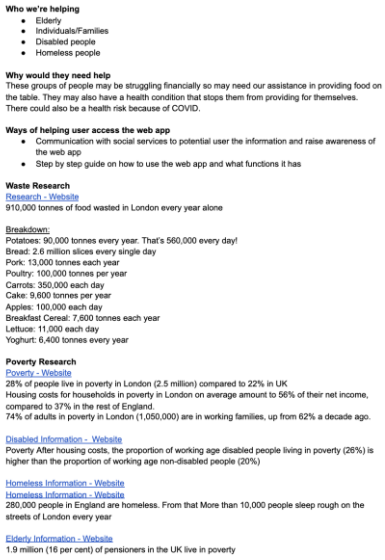
Donate Money · Issue #7 · BrunelCS/cs2001-2020_21-group10 (no date). Available at: https://github.com/BrunelCS/cs2001-2020_21-group10/issues/7 (Accessed: March 1, 2021).

Spring Boot Token based Authentication with Spring Security & JWT - BezKoder (no date). Available at: https://bezkoder.com/spring-boot-jwt-authentication/ (Accessed: March 1, 2021).

Appendix A Relevant Additional Material

A.1 Appendix Section 1

Research Screenshot for zero-hunger ideation:



Who we're helping
- Elderly
- Individuals/Families
- Disabled people
- Homeless people

Why would they need help
These groups of people may be struggling financially so may need our assistance in providing food on the table. They may also have a health condition that stops them from providing for themselves. There could also be a health risk because of COVID.

Ways of helping user access the web app
- Communication with social services to potential user the information and raise awareness of the web app
- Step by step guide on how to use the web app and what functions it has

Waste Research
Research - Website
910,000 tonnes of food wasted in London every year alone

Breakdown:
Potatoes: 90,000 tonnes every year. That's 560,000 every day!
Bread: 2.6 million slices every single day
Pork: 13,000 tonnes each year
Poultry: 100,000 tonnes per year
Carrots: 350,000 each day
Cake: 9,600 tonnes per year
Apples: 100,000 each day
Breakfast Cereal: 7,600 tonnes each year
Lettuce: 11,000 each day
Yoghurt: 6,400 tonnes every year

Poverty Research
Poverty - Website
28% of people live in poverty in London (2.5 million) compared to 22% in UK
Housing costs for households in poverty in London on average amount to 56% of their net income, compared to 37% in the rest of England.
74% of adults in poverty in London (1,050,000) are in working families, up from 62% a decade ago.

Disabled Information - Website
Poverty After housing costs, the proportion of working age disabled people living in poverty (26%) is higher than the proportion of working age non-disabled people (20%)

Homeless Information - Website
Homeless Information - Website
280,000 people in England are homeless. From that More than 10,000 people sleep rough on the streets of London every year

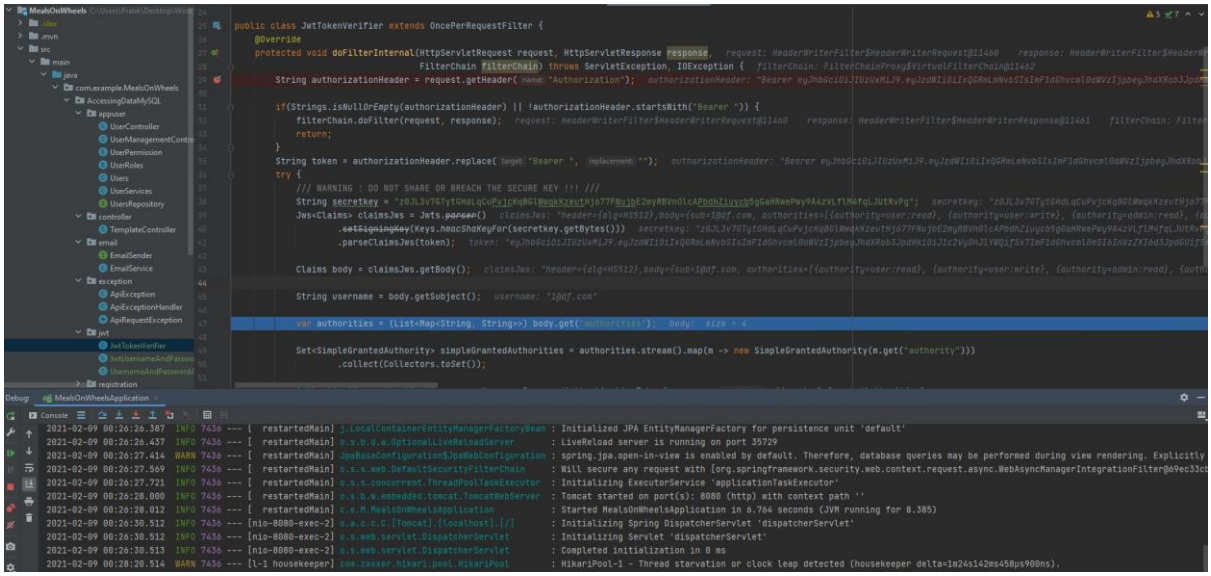
Elderly Information - Website
1.9 million (16 per cent) of pensioners in the UK live in poverty

A.2 Appendix Section 2

Token Verifier Class with Internal Filters:

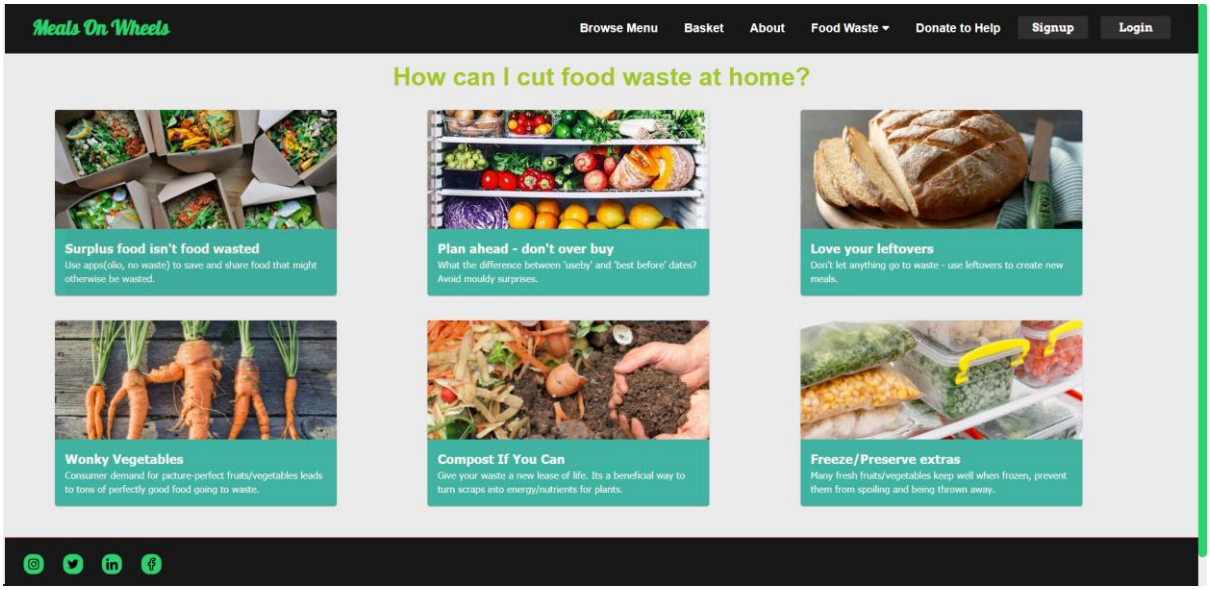
```
public JwtTokenVerifier(JwtConfig jwtConfig, SecretKey secretKey) {
    this.jwtConfig = jwtConfig;
    this.secretKey = secretKey;
}

@Override
protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response,
                                FilterChain filterChain) throws ServletException, IOException {
    String authorizationHeader = request.getHeader(jwtConfig.getAuthorizationHeader());

    // Completed : Could add validation to check if token is already valid by storing some of them.

    if(Strings.isNullOrEmpty(authorizationHeader) || !authorizationHeader.startsWith(jwtConfig.getTokenPrefix())) {
        filterChain.doFilter(request, response);
        return;
    }
    String token = authorizationHeader.replace(jwtConfig.getTokenPrefix(), replacement: "");
    try {
        Jws<Claims> claimsJws = Jwts.parser()
                .setSigningKey(secretKey)
                .parseClaimsJws(token);

        Claims body = claimsJws.getBody();

        String username = body.getSubject();

        var authorities = (List<Map<String, String>>) body.get("authorities");

        Set<SimpleGrantedAuthority> simpleGrantedAuthorities = authorities.stream().map(m -> new SimpleGrantedAuthority(m.get("authority")))
                .collect(Collectors.toSet());

        Authentication authentication = new UsernamePasswordAuthenticationToken(username, credentials: null, simpleGrantedAuthorities);

        SecurityContextHolder.getContext().setAuthentication(authentication);

    } catch(JwtException e) {
        throw new IllegalStateException(String.format("Token: %s cannot be trusted.", token));
        // TODO Carryout A New Exception Handling for IllegalStateException.
    }

    filterChain.doFilter(request, response);
}
```

Debugging TokenVerification:



A.3 Appendix Section 3

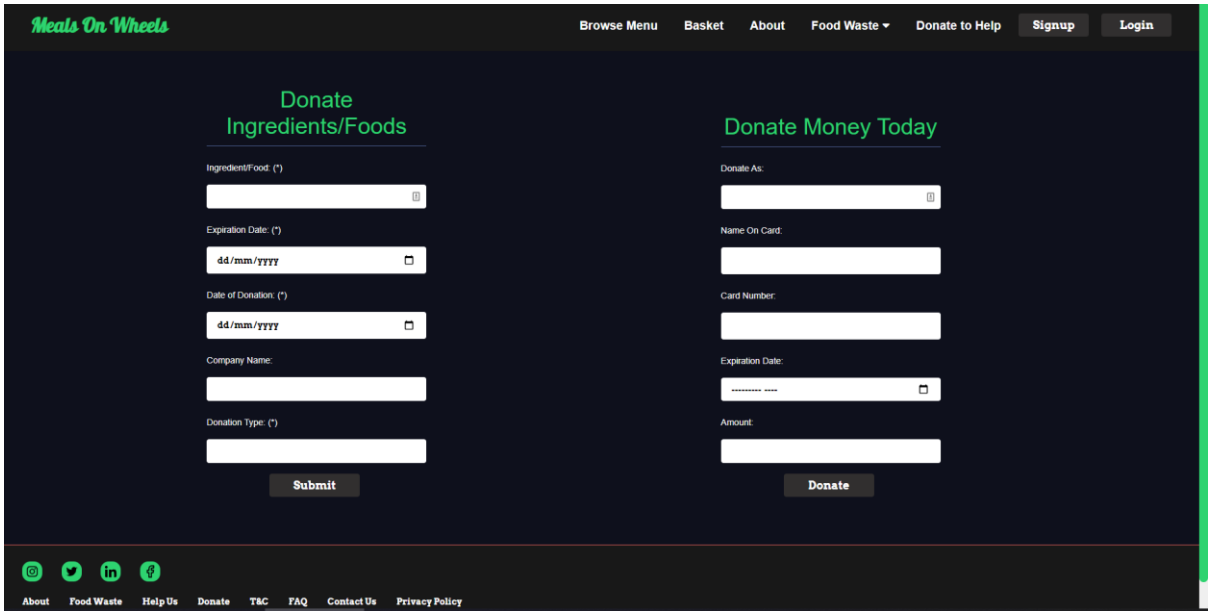
Created “How to Contribute” page:



Created Food Waste Facts Page:

Created a custom navigation bar (after login):

Created a new donate money form on the help page:


Meals On Wheels
Browse Menu
Basket
About
Food Waste
Donate to Help
Signup
Login
Donate Ingredients/Foods
Ingredient/Food: (*)
Expiration Date: (*)
dd/mm/yyyy
Date of Donation: (*)
dd/mm/yyyy
Company Name:
Donation Type: (*)
Submit
Donate Money Today
Donate As:
Name On Card:
Card Number:
Expiration Date:
Amount:
Donate
About
Food Waste
Help Us
Donate
T&C
FAQ
Contact Us
Privacy Policy