AI-capable web-based Application for the Continuous Food Safety Training and Assessment of Chefs.

Jose Leonardo Wong

Table of Contents

Introduction

This project proposes developing a web-based application integrated with the OpenAI GPT-4 Natural Language model API to address the need for continuous food safety training in commercial kitchens. Ensuring food safety in restaurants is critical and involves training chefs and maintaining training records. Existing software solutions focus on initial training and lack adaptability for ongoing training.

The project aims to develop a web application using the OpenAI GPT-4 Natural Language model to deliver ongoing food safety training and assessments tailored for professional chefs. This initiative aims to simplify the training and assessment process, ensuring its suitability for restaurant environments. The web-based application employs Natural Language Processing (NLP) via the OpenAI GPT-4 API to automatically create personalised quizzes and case scenarios for food safety training. Trainers can easily review and modify quizzes, assign them to chefs, and promptly receive feedback.

The project is motivated by the need to simplify continuous training for restaurant chefs, which is a legal obligation for restaurants. The application targets restaurant managers, head chefs, and trainers responsible for chef training. Chefs and kitchen staff benefit from enhanced food safety awareness, and customers in restaurants using the application benefit from heightened food safety measures.

It is important to note that the project's scope is predominantly focused on software development. While it exploits the capabilities of the AI model, it does not aim to explore the internal architecture or parameters of the natural language model. Instead, the project explores mechanisms to enhance the accuracy and relevance of the generated content by employing techniques such as prompt engineering and integrating relevant data retrieval systems and a food safety corpus of knowledge.

From the software development perspective, in addition to integrating the application with the OpenAI API, the project aims to follow the Agile software development methodology and use rigorous testing and professional project management tools to deliver the objective of an application that fulfils the functional requirements, including a user-friendly graphic interface. Additionally, the application will be deployed using a cloud platform.

The report consists of six chapters. Chapter 1, "Literature Review," explores Artificial Intelligence (AI) and Natural Language Processing (NLP), focusing on content accuracy and relevance, similar products, and food safety laws. Chapter 2, "Methodology," outlines the methods used for research and development. Chapter 3, "Specifications and Design," details the application's technical requirements and design principles. Chapter 4, "Development and Implementation," describes the development process through seven Agile sprints, noting progress and achievements. Chapter 5, "Testing and Results," covers testing strategies, including accessibility, markup, penetration, and cross-browser tests, and analyses the outcomes and user feedback. Chapter 6, "Critical Evaluation", evaluates the project against its goals, discusses deviations from the plan, assesses strengths and weaknesses, and reflects on lessons learned. Lastly, a “Conclusions” section compares the developed application against the state of the art, suggests improvements, and addresses legal and ethical considerations.

Chapter 1. Literature Review

This literature review provides a background for developing a web application with natural language generative capabilities. It mainly focuses on the current landscape of Artificial Intelligence (AI), particularly Natural Language Processing (NLP), the assurance of accuracy and relevance of AI-generated content and a selection of sources, including similar products. Additionally, the review considers sources discussing food safety compliance and legislation. This approach ensures a comprehensive understanding of the field and supports the development of the proposed web-based application utilising the OpenAI GPT-4 Natural Language Processing (NLP)

1.1. Natural Language Processing

Natural Language Processing (NLP) is a branch of Artificial Intelligence that focuses on enabling computers to understand and produce human language. A significant aspect of NLP is Natural Language Generation (NLG). NLG is concerned with allowing machines to generate coherent and contextually relevant text autonomously, simulating how humans create language. This capability represents an essential advancement in the field of NLP and is crucial for applications implementing automated content creation (Bhoi et al., 2023).

1.1.1. Artificial Intelligence and Natural Language Processing

Artificial Intelligence (AI) is a field within computer science that enables machines to perform rational actions similar to human decision-making, taking the best possible action in any situation. Achieving this involves complex challenges. These include accurately modelling and interpreting the environment, effectively representing information, engaging in complex reasoning under uncertainty, and continuously learning and adapting from experience. These aspects underscore the nature of AI and its need for advanced research in perception, representation, reasoning, and learning to create intelligent systems (Russell & Norvig, 2022).

As mentioned before, learning and adapting are crucial elements to characterise a system as intelligent. This is where Machine Learning (ML), a subset of AI, plays a crucial role. ML uses various other disciplines, such as statistics, neurobiology, and psychology. It focuses on enabling computers to learn and improve from experience, aiming to achieve intelligent behaviour without explicit programming for each task. ML's effectiveness is evident in various applications where it is applied, including speech recognition and fraud detection, where it often performs better than humans (Mitchell, 1997).

Following the exploration of Machine Learning (ML) as a means for machines to learn and adapt, Generative AI represents an advanced evolution of this concept, where the focus shifts from understanding and analysing data to creating new, realistic instances of it. Central to Generative AI are generative models that, unlike their discriminative counterparts, do not just categorise or label data but learn to emulate and generate data akin to the original set. These models, operating on principles of probability and incorporating elements of randomness, excel in tasks like text generation. For instance, a generative model trained on a vast corpus of literature can produce novel textual content, such as stories or articles, that mirrors human writing in style and substance. This capability of Generative AI to innovate by synthesising new data, be it text, images, or other forms, marks a significant leap in the journey towards more creative and autonomous AI systems (Foster, 2019).

Building on the concepts of AI and ML, Deep Learning (DL) emerges as a specialised subset of ML. It employs Deep Neural Networks, a multi-layered structure of connected nodes that mimic the structure of a human neuron to execute intricate computations in data.

Deep Learning has become crucial in applications like visual object recognition, machine translation, and speech synthesis due to its ability to handle tasks involving large amounts of data and complex patterns.

The field of Natural Language Processing (NLP) has greatly benefited from these advances in AI. For instance, Generative AI has revolutionised the field, enabling computers to produce human-like text and synthesise information from large datasets to create original content, such as news reports and creative writing. This evolution in Generative AI allows for creating more sophisticated applications that can produce human language (Song et al., 2019). Similarly, DL has allowed the creation of language models because neural networks are particularly well-suited for handling and processing language data due to their ability to learn complex patterns and relationships within the data (Russell & Norvig, 2022).

1.1.2. Enhancing Reliability and Accuracy of Large Language Model Output

As demonstrated in the previous section, the evolution of Large Language Models (LLMs) like GPT-4 has resulted in significant improvements in achieving human-like accurately generated language. However, these models have limitations, as they can sometimes produce factually incorrect responses or "hallucinations." Moreover, the indeterministic nature of neural networks and particularly large language modes such as GPT-4, is a situation that must be addressed when the generated output is expected to have precision and accuracy, particularly in safety-critical when integrating the models into safety-critical applications (Liu et al., 2023). In developing our web-based application for chef training in food safety, ensuring the accuracy of content is crucial due to its safety-critical nature.

Fine-tuning GPT models via the API makes it possible to tailor the AI's responses to specific applications and domains, such as food safety. This process involves training the model on a dataset that aligns with the desired task, enabling it to learn specific patterns. For instance, fine-tuning using food safety legislation and a large, labelled dataset of quizzes could improve performance, reducing errors and ensuring the generated content adheres to specific requirements. However, it is essential to note that the possibility of doing this with the GPT-4 model is at an experimental stage, and access is limited when writing this literature review. Moreover, OpenAI recommends exploring prompt engineering as a preliminary step before considering fine-tuning, as it can often yield satisfactory results without the need for fine-tuning (OpenAI, 2023). Fine-tuning the model itself is beyond the scope of this project. However, our project aims to incorporate the required software code into the application, making it possible to explore fine-tuning possibilities in the future, for instance, by using gathered user feedback.

Based on the evidence and to enhance the accuracy and relevance of generated content in our application, we propose a Retrieval-Augmented Language Model (RALM) approach to address this. This approach combines a language model like GPT with a retrieval system—a tool that fetches relevant, up-to-date information from food safety databases based on specific queries. We align the content with current food safety standards by using GPT for initial content generation and enhancing it with accurate, external data from the retrieval system. Recent studies in this respect demonstrate that this straightforward method significantly improves content accuracy even through API integration without manipulating the natural language model (Ram et al., 2023). This illustrates how our application could produce factually correct, engaging, and up-to-date quizzes by appending relevant, verified information from external food safety databases to the prompts given to GPT-4. Consequently, we propose implementing this retrieval system approach using the Elasticsearch engine and a corpus of relevant food safety good practices literature and legislation to integrate with our application and enhance the accuracy and relevance of the content generated.

In addition to using a retrieval system to enhance the accuracy and relevance of the content generated by our application, we proposed employing prompt engineering, a strategic approach in natural language processing. Prompt engineering involves carving prompts and instructions to utilise large language models (LLMs) effectively. Within this approach, we utilise 'prompt patterns,' which provide a flexible framework for customising LLM outputs. Similar to software patterns, these patterns ensure that our quizzes meet the highest food safety standards. For example, the 'Question Refinement Pattern' refines queries, such as 'create a quiz about safe food temperatures,' to specific details like 'chicken internal cooking temperature of 75 degrees Celsius'. Research has shown that this prompt engineering approach, including the use of techniques like 'zero-shot' (incorporating an example in the prompt) and 'few-shot' (providing multiple examples), when combined with advanced prompt patterns like 'Question Refinement' and 'Persona' (associating a role with the prompt), are highly effective in the domain of software requirements elucidation using generative AI and can help to identify reusable best-performing prompts. We consider that our project can benefit from the same advantages, for instance, avoiding resource-intensive supervised learning techniques depending on large amounts of labelled data to enhance the generated output accuracy and relevance (Ronanki, et al., 2023).

1.2. Food Safety Legislation and Best Practices

To develop GPT prompts that are relevant and accurate and to assess generated content quality, it is crucial to consider and integrate relevant legislation and establish good practices. This ensures that the training quizzes and generated content align with the highest food safety and hygiene standards. In the next lines, we present the most important pieces of legislation and best practices that will need to be considered for use as information for the prompt engineering process as well as a corpus of knowledge for the proposed retrieval system integration.

The Hazard Analysis and Critical Control Points (HACCP) standard (U.S. Food & Drug Administration, 2022), introduced in the late 1980s by the U.S. National Advisory Committee on the Microbiological Criteria for Food (NACMCF), became the world's first international HACCP standard, adopted by the Codex Alimentarius Commission—an international body under the FAO and WHO whose primary purpose is to develop international food standards, guidelines, and codes of practice to ensure the safety, quality, and fairness of food trade on a global scale. HACCP is a hazard prevention tool used throughout the food chain. The HACCP System is a globally recognised food safety system based on the HACCP principles that identify and control hazards during food production with a focus on prevention. The ISO 22000:2018 (International Organization for Standardization, 2018) international standard incorporates the principles of HACCP into a Food Safety Management System.

Regulation (E.C.) No 852/2004, dated 29 April 2004 (European Parliament and Council of the European Union), establishes hygiene regulations for various food businesses, encompassing production, processing, distribution, and sales. It outlines food safety and hygiene standards, emphasising the application of Hazard Analysis and Critical Control Points (HACCP) principles to safeguard food safety. The regulation places the primary responsibility for food safety on the food business operator. Food business operators must take suitable measures to ensure their staff members handling foodstuffs receive health risks and food hygiene training. After Brexit, EU law was incorporated into the U.K. The European Union (Withdrawal) Act 2018, which came into force in the U.K. on 31 January 2020, preserved existing E.U. laws as they were at the end of the Brexit transition period (31 December 2020) and converted them into U.K. law.

The Food Safety Act of 1990 (Great Britain, 1990) is a significant piece of legislation that establishes several provisions related to food safety, including food handling, storage, and distribution. It is designed to ensure that food intended for human consumption is safe, meets the necessary quality standards, and establishes the responsibilities of food businesses to ensure food safety.

The Food Standards Act 1999 (Great Britain, 1999) created the Food Standards Agency (FSA) in the United Kingdom. The FSA plays a vital role in setting food safety standards, conducting research, and providing information to the public regarding food safety. The Act provides the legal framework for regulating food safety and standards in the U.K.

The Food Safety and Hygiene (England) Regulations 2013 (Great Britain, 2013) are a set of regulations in England that pertain to food safety and hygiene standards. These regulations provide guidelines and requirements for food businesses to ensure the safety of foodstuffs and maintain hygiene practices. They cover various aspects, including establishing FSA powers, codes of recommended practice for food authorities, and compliance with Hygiene Regulations.

The Food Standards Agency 2021 Food Law Practice Guide (England) (Food Standards Agency, 2021) provides guidance and information on food standards and practices in England, offering recommendations and instructions for various aspects of food safety and hygiene. Chapter 4 of the document mandates that food businesses establish procedures to control food safety hazards and integrate them with documentation and record-keeping. Staff must have the necessary skills, follow safety procedures, receive proportional training, and undergo on-the-job training as needed.

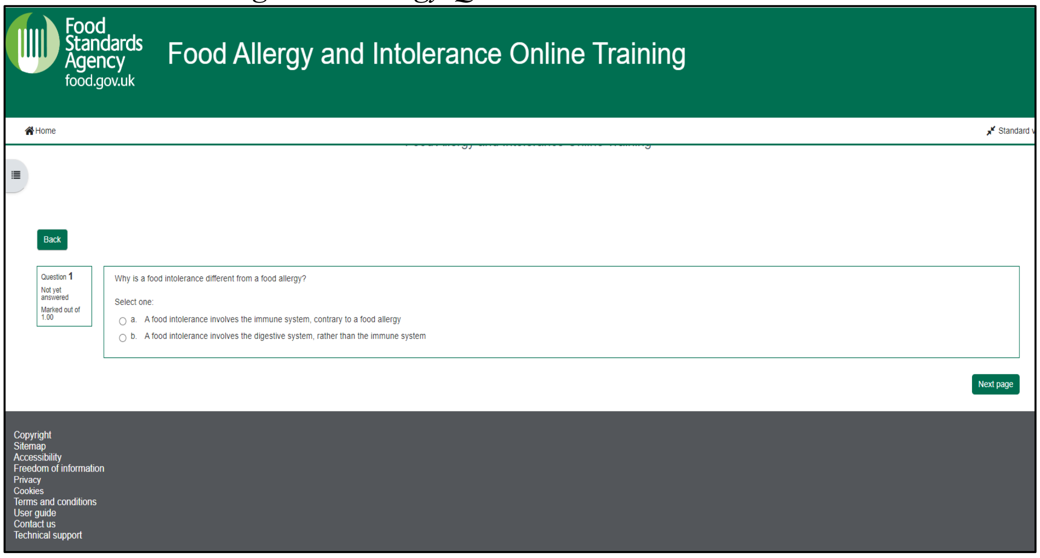
The Safer Food, Better Business package (Food Standards Agency, 2021) is a document to assist small- and medium-sized food businesses in adhering to food safety regulations. This practical guide offers tailored advice on food safety management, hygiene, and record-keeping for various catering establishments. Its purpose is to help caterers establish and maintain effective food safety procedures to ensure the safety of the food they provide to customers.

1.3. Similar Applications

The official Food Standards Agency online training is limited to allergies and food intolerances. The training module includes quizzes consisting of multiple-choice questions about allergen awareness concepts. As seen in Figure 1, the quizzes do not generate case scenarios. Upon completion of the test, it provides a score and details of wrong and right answers, but no detailed feedback or explanation is given. The content repeats every time the quiz is taken, focusing on initial training, not ongoing training.

Figure 1

FSA Online Training Food Allergy Quiz



Food Standards Agency food.gov.uk

Food Allergy and Intolerance Online Training

Home

Standard v

Back

Question 1
Not yet answered
Marked out of 1.00

Why is a food intolerance different from a food allergy?

Select one:

- ○ a. A food intolerance involves the immune system, contrary to a food allergy
- ○ b. A food intolerance involves the digestive system, rather than the immune system

Next page

Copyright
Sitemap
Accessibility
Freedom of information
Privacy
Cookies
Terms and conditions
User guide
Contact us
Technical support

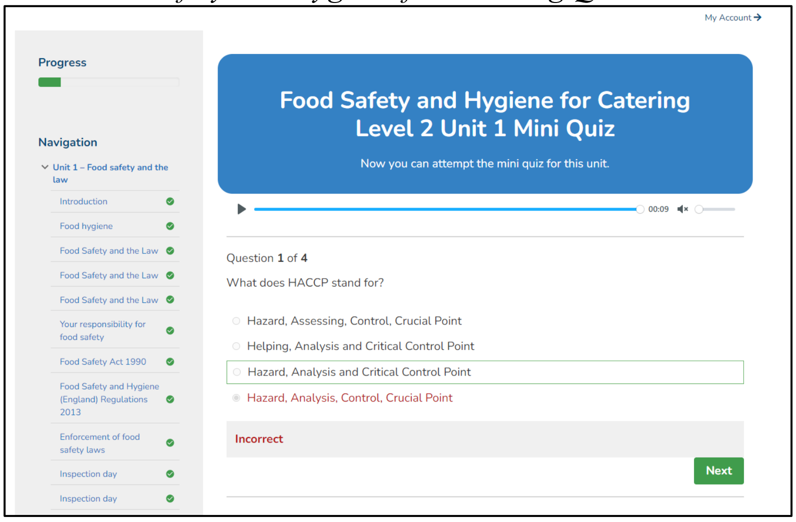
Note. Demonstration of a question part of a quiz.

Generated using the Food Allergy and Intolerance Training (https://allergytraining.food.gov.uk/)

CPD is the leading provider of online training in food safety for restaurants. The training modules are extensive content covering all aspects of food safety. As seen in Figure 2, each module has a quiz at the end. The quizzes consist of multiple-choice questions related to food safety concepts. They do not use case scenarios as part of the quiz. Upon completion of the test, it provides a score and details of wrong and right answers, but no detailed feedback or explanation is given. The content repeats every time the quiz is taken, focusing on initial training, not ongoing training.

Figure 2

CPD Food Safety and Hygiene for Catering Quiz



Food Safety and Hygiene for Catering Level 2 Unit 1 Mini Quiz

Now you can attempt the mini quiz for this unit.

Question 1 of 4

What does HACCP stand for?

- Hazard, Assessing, Control, Crucial Point
- Helping, Analysis and Critical Control Point
- Hazard, Analysis and Critical Control Point
- Hazard, Analysis, Control, Crucial Point

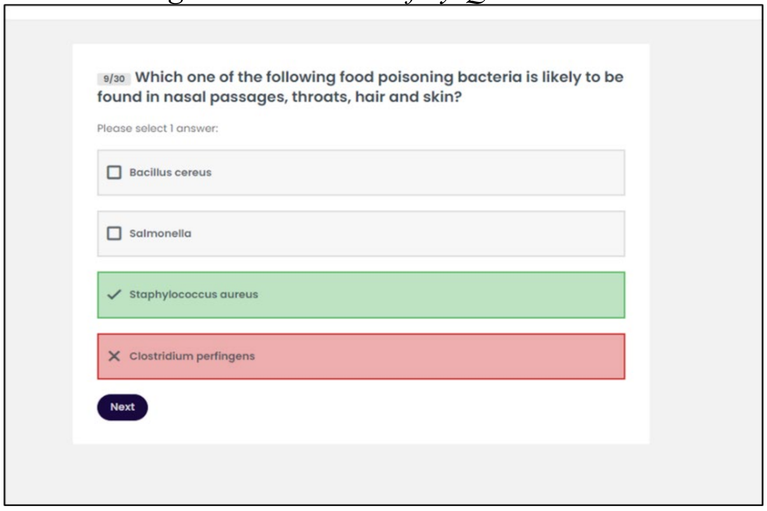
Incorrect

Note. Demonstration of a question part of a CPD food safety quiz. Generated using the CPD Food Safety and Hygiene for Catering Quiz (https://cpdonline.co.uk/course/food-safety-hygiene-catering-level-2/)

Virtual College is another provider of food safety online training. Like the previous examples, the training modules are followed by quizzes consisting of multiple-choice questions related to food safety concepts. As seen in Figure 3, the quizzes do not use case scenarios. Upon completion of the test, it provides a score and details of wrong and right answers, but no detailed feedback or explanation is given. The content repeats every time the quiz is taken, focusing on initial training, not ongoing training.

Figure 3

Virtual College Online Food Safety Quiz



9/30 Which one of the following food poisoning bacteria is likely to be found in nasal passages, throats, hair and skin?

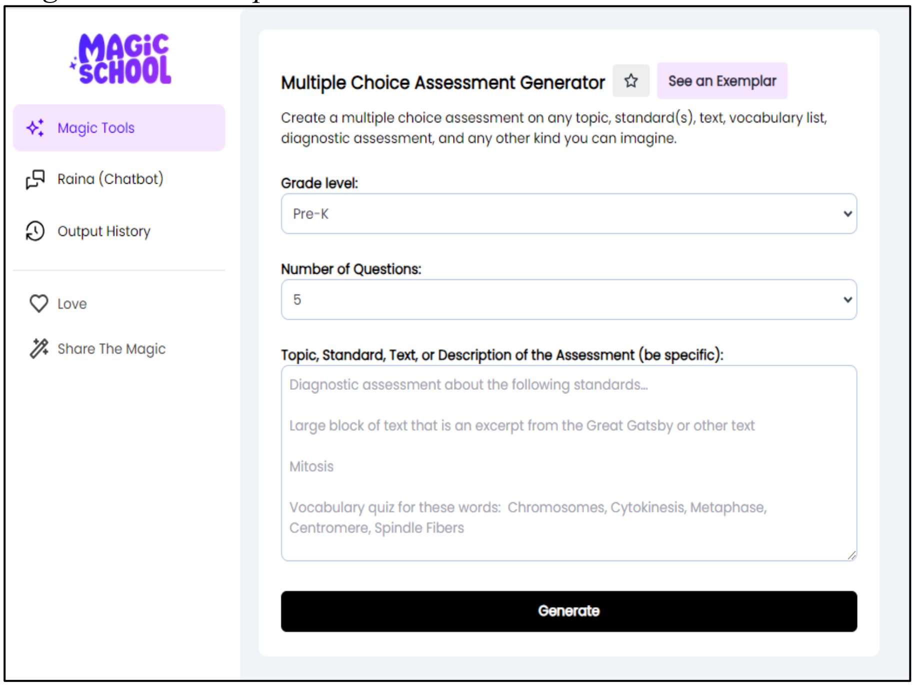
Please select 1 answer:

- ☐ Bacillus cereus
- ☐ Salmonella
- ✓ Staphylococcus aureus
- ✗ Clostridium perfingens

Note. Demonstration of a question part of a Virtual College Food Safety Training. Generated using the College Online Food Safety Quiz (https://www.virtual-college.co.uk/food-hygiene-training-courses

Magic School is an online application that allows teachers to generate multiple teaching resources. As demonstrated in Figure 4, the application is AI-powered. It can generate multiple-choice quizzes for schools and colleges based on a topic and the required number of questions the user enters. The generated pieces of assessment are unique on every request. As seen in Figure 5, the application can be prompted to generate food safety-related multiple-choice quizzes, but there is no generation of case scenarios. The generated quizzes can be used to generate rubrics, which are based on specific criteria entered by the user. However, this requires an additional process, and the rubric is not generated along with the multiple-choice quiz. The application targets educational institutions, not the restaurant industry.

Figure 4
Magic School Multiple Choice Assessment Generator


Multiple Choice Assessment Generator

Create a multiple choice assessment on any topic, standard(s), text, vocabulary list, diagnostic assessment, and any other kind you can imagine.

Grade level: Pre-K

Number of Questions: 5

Topic, Standard, Text, or Description of the Assessment (be specific):

Diagnostic assessment about the following standards...

Large block of text that is an excerpt from the Great Gatsby or other text

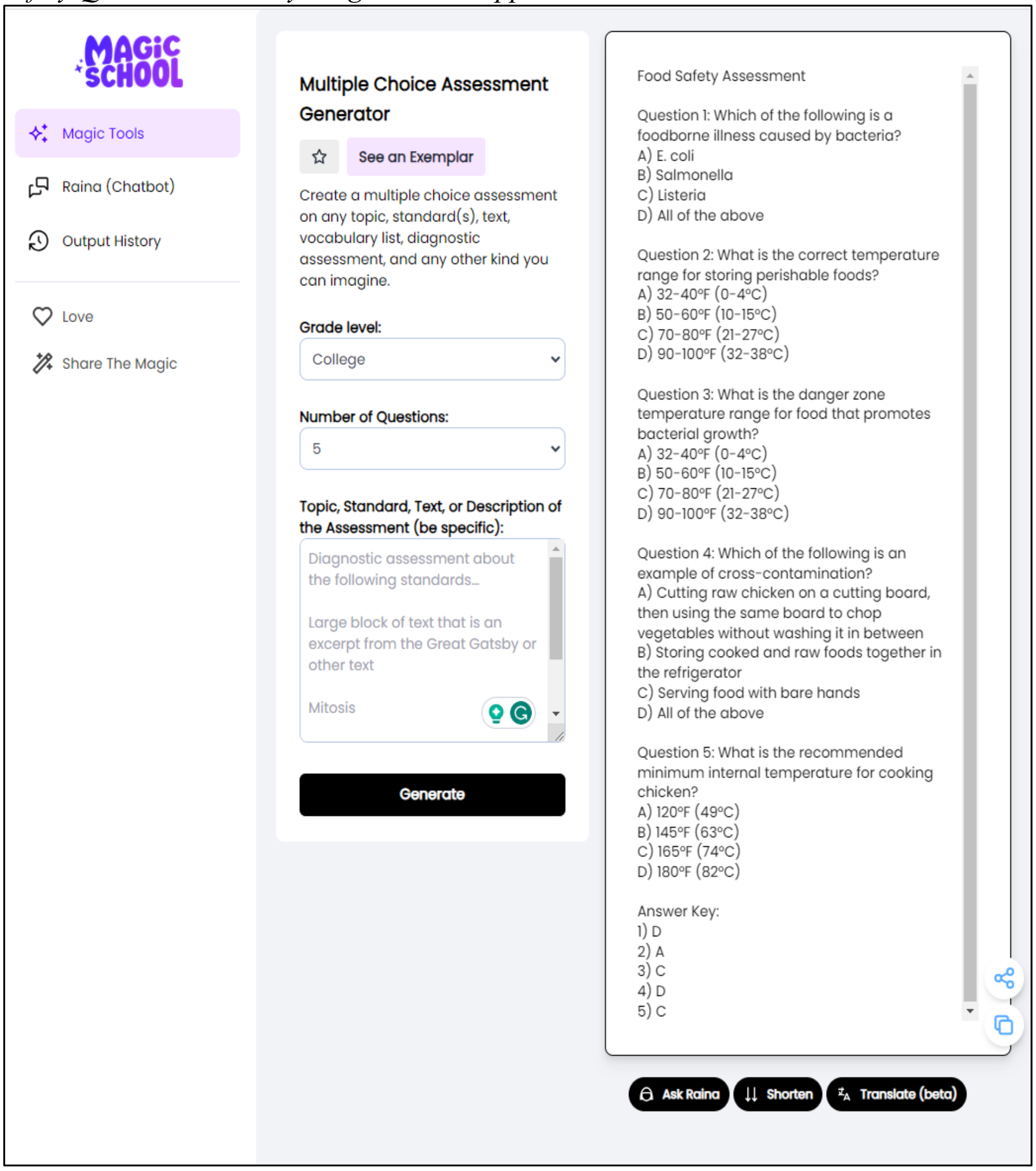
Mitosis

Vocabulary quiz for these words: Chromosomes, Cytokinesis, Metaphase, Centromere, Spindle Fibers

Note. Demonstration of Quiz Generation using Magic School. Generated using the Magic School Application

Figure 5

Safety Quiz Generated by Magic School Application



Multiple Choice Assessment Generator

Create a multiple choice assessment on any topic, standard(s), text, vocabulary list, diagnostic assessment, and any other kind you can imagine.

Grade level: College

Number of Questions: 5

Topic, Standard, Text, or Description of the Assessment (be specific):

Diagnostic assessment about the following standards...

Large block of text that is an excerpt from the Great Gatsby or other text

Mitosis

Food Safety Assessment

Question 1: Which of the following is a foodborne illness caused by bacteria?
A) E. coli
B) Salmonella
C) Listeria
D) All of the above

Question 2: What is the correct temperature range for storing perishable foods?
A) 32-40°F (0-4°C)
B) 50-60°F (10-15°C)
C) 70-80°F (21-27°C)
D) 90-100°F (32-38°C)

Question 3: What is the danger zone temperature range for food that promotes bacterial growth?
A) 32-40°F (0-4°C)
B) 50-60°F (10-15°C)
C) 70-80°F (21-27°C)
D) 90-100°F (32-38°C)

Question 4: Which of the following is an example of cross-contamination?
A) Cutting raw chicken on a cutting board, then using the same board to chop vegetables without washing it in between
B) Storing cooked and raw foods together in the refrigerator
C) Serving food with bare hands
D) All of the above

Question 5: What is the recommended minimum internal temperature for cooking chicken?
A) 120°F (49°C)
B) 145°F (63°C)
C) 165°F (74°C)
D) 180°F (82°C)

Answer Key:
1) D
2) A
3) C
4) D
5) C

Note. Demonstration of the Generation of a Food Safety Quiz using the Magic School.
Generated using the Magic School Application (https://www.magicschool.ai/)

1.4. Conclusions

Our literature review has allowed us to explore the possibilities of using generative NLP to provide our application with the ability to generate dynamic quizzes that can be used for chefs' ongoing food safety training. One of the main findings is that using NLP has limitations, and generated content can be erroneous or generate “hallucinations” due to the nondeterministic nature of neural networks. Consequently, we have realised that the state of the art of NLP does not allow for prescinding of the human component, and we need to, in addition to ensuring that a knowledgeable person revises the generated quizzes, explore mechanisms to ensure the generated content of our application is safe and accurate.

An important realisation has been that there are mechanisms such as prompt engineering and the use of retrieval systems to improve the accuracy and relevance of the generated content, which is a critical aspect of this project. Similarly, we have found that there are opportunities to expand this project to explore the possibilities of fine-tuning the GPT-4 model using the OpenAI API. Moreover, integrating our application with open-source natural language models trained for fact-checking, for instance, could increase the quality of the generated content and safety net to avoid providing erroneous training. This combined approach of using a natural language model with a retrieval system and appropriate prompt engineering techniques and patterns ensures that the content produced by our application not only meets but exceeds the highest standards of accuracy and relevance for effective chef training in food safety.

Similarly, we have found that several applications in the market offer online food safety training. Still, none are focused on continuous training and do not have A.I. capabilities or produce changing content over time. Although some products in the market are designed to produce quizzes and teaching resources using generative NLP, these applications are not focused on the restaurant industry and do not use a case-scenario approach. These findings open a possibility to integrate our dynamic quiz generation application into other food safety online training applications to provide them with the same generative NLP quiz generation capabilities or alternatively escalate our application to provide quizzes and the delivery of training modules.

Chapter 2. Methodology

2.1. Software Development Methodology

Our application's development process will require substantial experimentation, trial, and error to arrive at an appropriate set of prompts to generate content via the API. Choosing the appropriate software methodology is crucial for our project. We have chosen the Agile software development methodology. Agile offers an iterative approach, adaptability to changing requirements, flexibility, and an emphasis on client collaboration. In contrast, other plan-based methodologies, such as Waterfall, emphasise planning and a rigid sequence of stages. Agile is ideal for this project due to its responsiveness to changing needs, particularly in integrating API into the application and assessing its performance to generate suitable content. It may require adapting different approaches and trying different prompt patterns. Agile will allow us to conduct frequent testing, provide permanent user feedback, and identify potential shortcomings in performance early. This approach will also allow us to produce a minimum viable product and build more refined versions as we progress (Demirag, Demirkol Ozturk, & Unal, 2023).

2.2. Research Methodology

A mixed qualitative and quantitative research approach will be utilised to execute the project. The qualitative approach will include a literature review of the evolution of AI, the state of the art of natural language processing, and existing applications similar to the one proposed in this project. Similarly, we will interview potential application users, such as Head Chefs and kitchen staff, to comprehend their needs and expectations. Moreover, we will perform interviews during the testing stage after deploying the application to assess the user's reaction. Regarding the quantitative analysis, we will use statistical analysis applied to the data obtained during the interviews as well as the application performance in terms of metrics obtained from the deployment tools in the production environment as well as comparing the accuracy of generated content with relevant food safety legislation and best practices documentation.

2.3. Programming Languages and Tools

We will use a combination of software development tools to deliver our application. This selection will include the Symfony PHP framework for the back end. Symfony will allow us to simplify the development process by streamlining the integration of the diverse entities in our code, such as a user and a quiz, with our database and the GPT-4 API. Additionally, Symfony will allow the application to be scalable in the future. For the database, we have chosen MySQL, a reliable, easy-to-use relational database management system perfect for handling structured data. For the front end, we will use Twig, Symfony's template engine and Bootstrap, a straightforward CSS framework with a wide selection of predesigned components.

For testing purposes, we will use PHPUnit, which integrates perfectly with Symfony and is the standard testing framework for PHP. Regarding project management, we have chosen Trello as a tool to help us track the progress of our application and schedule the iterations of our project in the context of Agile. Similarly, we will use Git, the most popular version control tool. Heroku, a cloud platform as a service (PaaS), is our selection of deployment tool in the cloud due to its simplicity and the possibility of scaling the deployment as needed in the future. Similarly, the open-source Elasticsearch will serve as the backbone of our retrieval system due to its capacity to index vast amounts of data from food safety databases and legislation and quickly retrieve relevant data, which will then be appended to the prompts to enhance the accuracy of the generated output.

Regarding the provision of natural language processing for the application, we have chosen the OpenAI GPT API particularly to integrate with the GPT-4 as we want to take advantage of the most advanced model. This integration will allow the application to generate engaging quizzes that dynamically change over time. Although there are several other options for natural language models, including open-source models such as LLaMA and BLOOM, our decision to choose GPT-4 is based on the streamlined possibilities of integration with the API and the evidence that GPT-4 vastly outperforms other open-source large language models in aspects such as commonsense reasoning and story comprehension and generation, which are critical for our application's ability to generate accurate quizzes containing realistic case scenarios (Naveed et al., 2023).

2.4. Design Patterns

Using proven structures for recurrent tasks will have a beneficial effect on our project delivery. Specifically, software design patterns such as the Model-View-Controller (MVC), enforced when using a framework such as Symfony, will allow our code to be more organised and manageable by separating business logic, control, and user interface concerns. Similarly, the Factory Pattern will be used to generate instances of different types of quizzes. Moreover, the Singleton pattern, commonly used to generate components that only need to have a single instance, will be used to connect with the database. (Gamma, Helm, Johnson, & Vlissides, 1995)

Chapter 3. Specifications and Design

The application is designed around the proposed core functionality of using OpenAI's GPT-4 API to incorporate its natural language processing capabilities into a website tool for professional chefs' food safety training. Integrating GPT-4's AI into this framework ensures the application can continuously provide varied content and assessment experiences.

Several factors influenced the choice of a web platform over a desktop or mobile application. First, web-based applications are accessible from any device with internet access without installing specific hardware or software, broadening the potential user base. This choice makes the maintenance process easier and ensures users can access the latest features and content. Additionally, the scalability of web applications allows flexibility in adjusting to variable demand and user load.

The overall application's design philosophy prioritised ease of use, maintenance, and the delivery of continuously evolving and dynamic educational content that could adapt to food safety training needs and potential changes in food safety standards, legislation, and best practices. The potential users' needs, findings, and feedback later ratified this philosophy and initial choices.

3.1. Requirements Elicitation

The functional requirements for the application were obtained through direct interactions with industry professionals, head chefs, and restaurant managers. Twenty professionals in the field were asked to provide continuous feedback and initially interviewed using an interview guide (Appendix E), were participants, were required consent to participate (Appendix C), briefed about the context and intention of the project and their expected participation (Appendix D) and asked to provide information about their roles in their companies, how they currently get involved in food safety training, the features they would expect to see implemented in the application, the food safety topics they would like to be considered for the generated assessment resources and their expectations regarding usability and interactivity.

As a result of these initial interviews, an initial overall set of functional and non-functional requirements was outlined and later refined during the implementation stage. As shown in Table 1, the functional requirements detail the specific actions and features the application must support, such as quiz generation and user management. As presented in Table 2, non-functional requirements represent the desired performance, usability, and system qualities.

Table 1
Overall Functional Requirements

Overall Functional Requirements
The system must support different types of users: Trainer and Chefs.
The system must automatically generate food safety quizzes.
The system must allow the customisation of quiz parameters to fit specific food safety topics: allergens, cross-contamination, food preparation and cooking methods, food storage methods, and cleaning and hygiene.
The system must enable the trainer to review and edit quizzes before assignment. The system must allow the trainer to assign quizzes to individual chefs or groups.

Table 1 (Continuation)
Overall Functional Requirements

Overall Functional Requirements
The system must maintain a record of quizzes assigned and completed, including dates, assigned chefs, feedback, and scores.
The system must automatically generate feedback for completed quizzes.

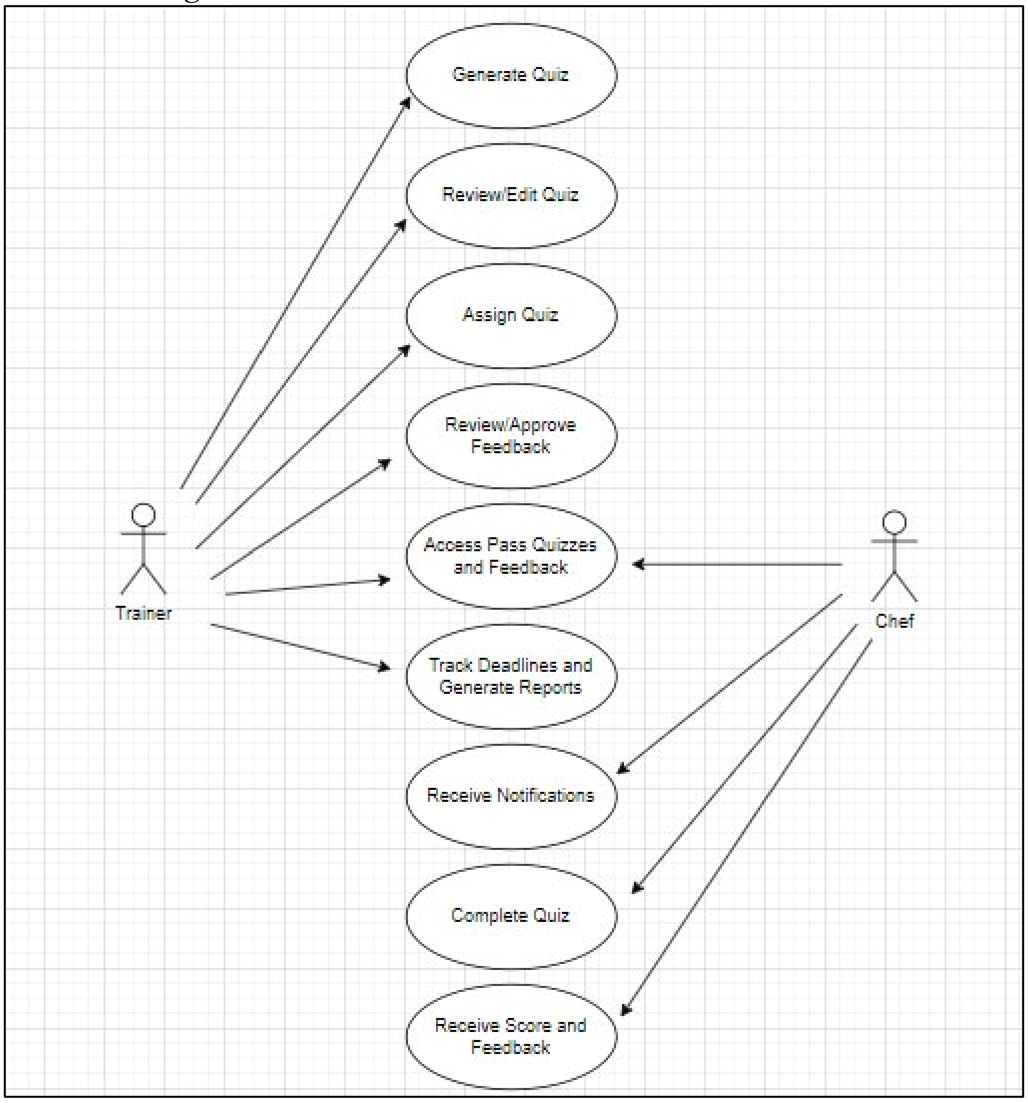
Table 2
Overall Non-Functional Requirements

Overall Non-Functional Requirements
The application must be user-friendly and intuitive.
The application must be accessible via standard web browsers without additional software installation.
The system must generate quizzes within a reasonable time to ensure efficient workflow.
The system must provide accurate outputs for quizzes and feedback.
The system must ensure the confidentiality of the users' data and comply with relevant data protection regulations.
The system must be usable through desktop computers and tablets with a screen size of at least 10 inches.

Consequently, the proposed system was structured to serve two main types of users: Trainers and Chefs. Trainers, including restaurant managers and head chefs, are responsible for overseeing food safety training and compliance. Chefs representing the learners who will engage with the quizzes to enhance their knowledge and adherence to food safety practices. The use-case diagram, depicted in Figure 10, visually maps the Food Safety Training Application's functionality, capturing the interactions between the system and these two principal user roles: Trainers and Chefs.

The use-case diagram serves two purposes: it provides a high-level view of the application's operations, illustrating the available actions linked to each role, and it acts as a connection between the defined functional requirements and the actual system use (Sommerville, 2016). The trainer is equipped with quiz generation, customisation, and management capabilities, while the chef interacts primarily through quiz completion and feedback review processes.

Figure 6
Use Case Diagram

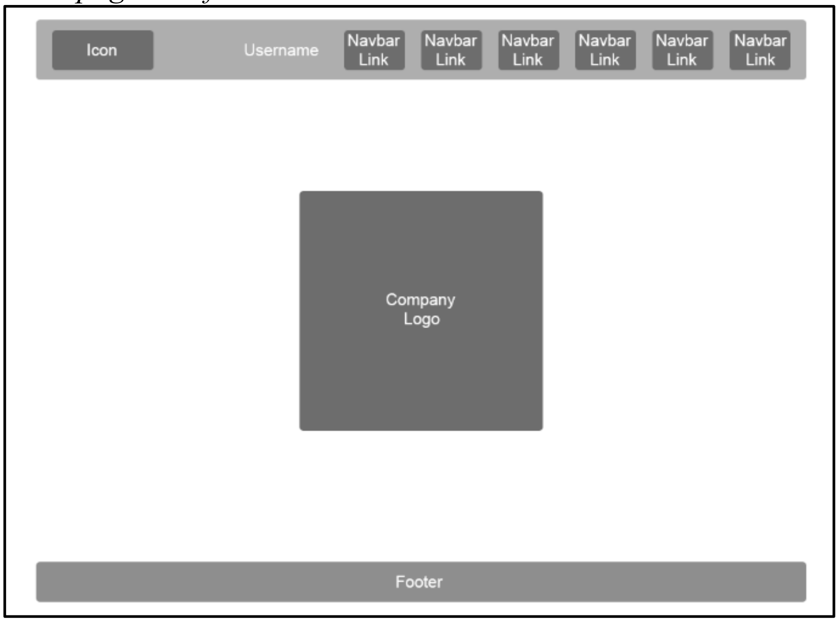


3.2. User Interface Design

The application's user interface was designed using wireframes to illustrate the proposed layout and interactive elements on each of the website's pages. This approach ensured the design adhered to the identified functional requirements. Additionally, the use of wireframes allowed for an early and clear visualisation of the user interface and, consequently, early evaluation and feedback from potential users.

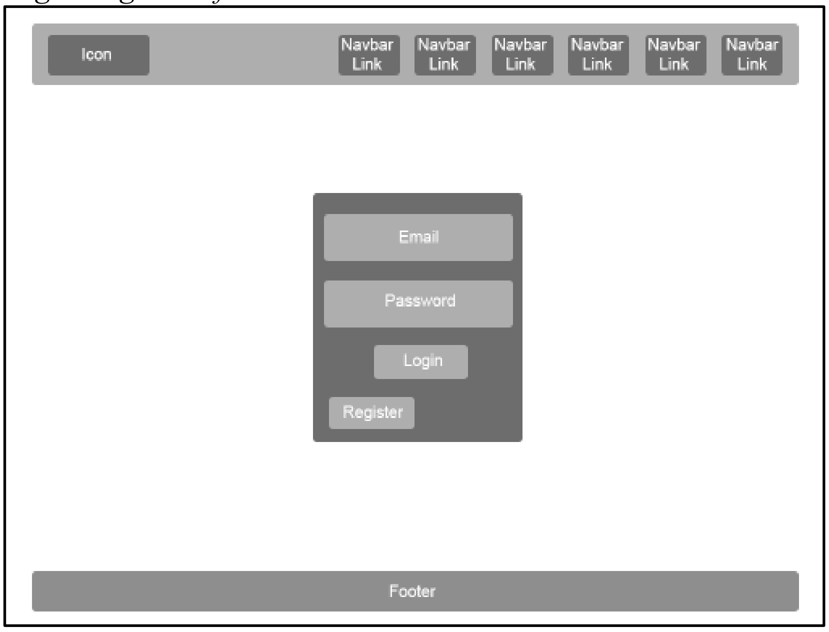
The homepage, as shown in Figure 7, was developed during the first sprint and is designed for simplicity. It presents the company's logo and a navigation bar. This approach ensures users are greeted with a clutter-free environment, easily identify the application's intended functionality, and have intuitive access to the other pages.

Figure 7
Homepage Wireframe



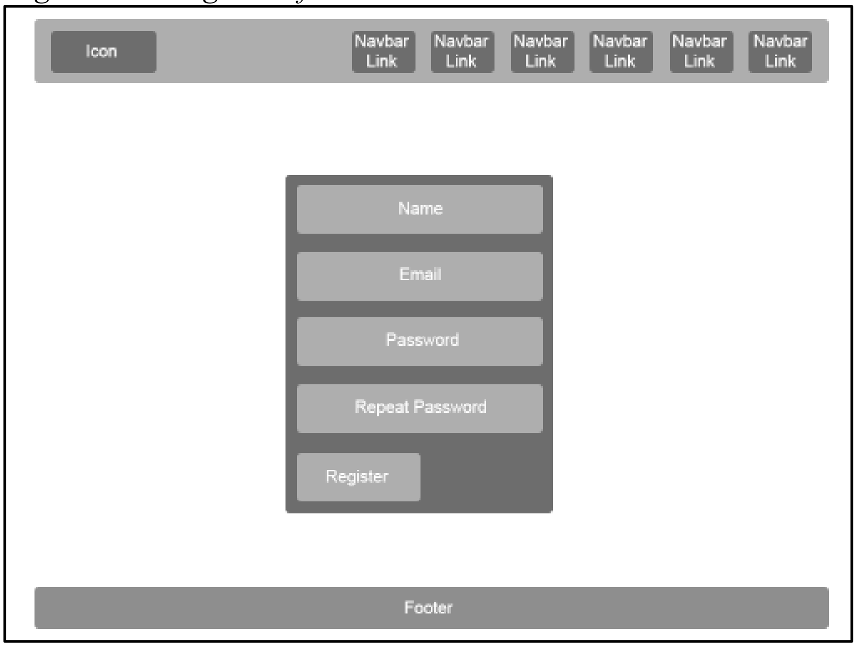
The "Login" page wireframe, designed at the beginning of the second sprint, focuses on user accessibility with a clean, unambiguous form for entering credentials, as displayed in Figure 8,

Figure 8
Login Page Wireframe



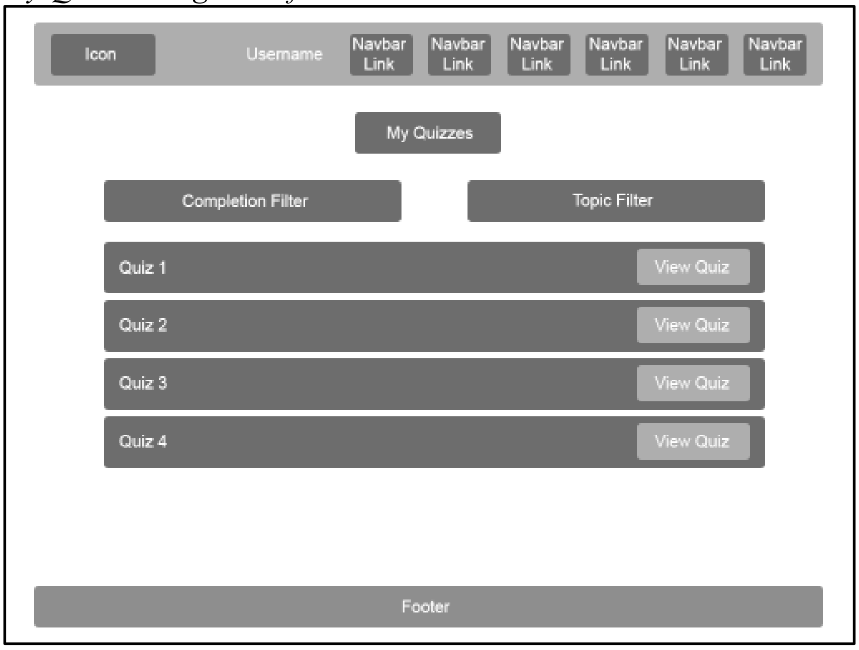
The registration page also designed in the second sprint, outlines a straightforward information collection form, as presented in Figure 9.

Figure 9
Registration Page Wireframe



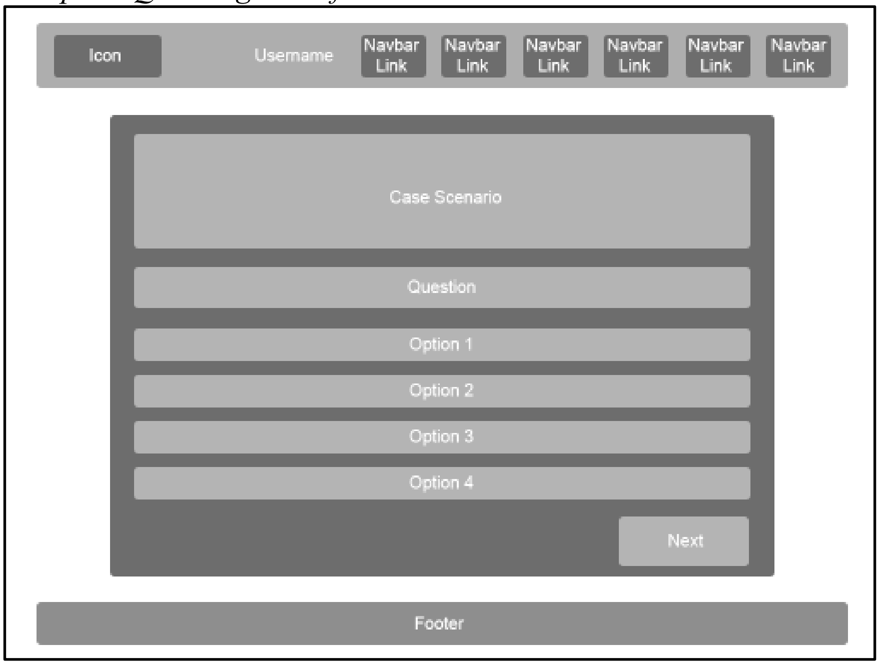
The 'My Quizzes' page, part of the design stage in the fifth sprint, is intended to display a list of quizzes assigned to a user alongside a button next to each quiz to either access the complete quiz page or view the details of a quiz previously completed, as shown in Figure 10

Figure 10
My Quizzes Page Wireframe



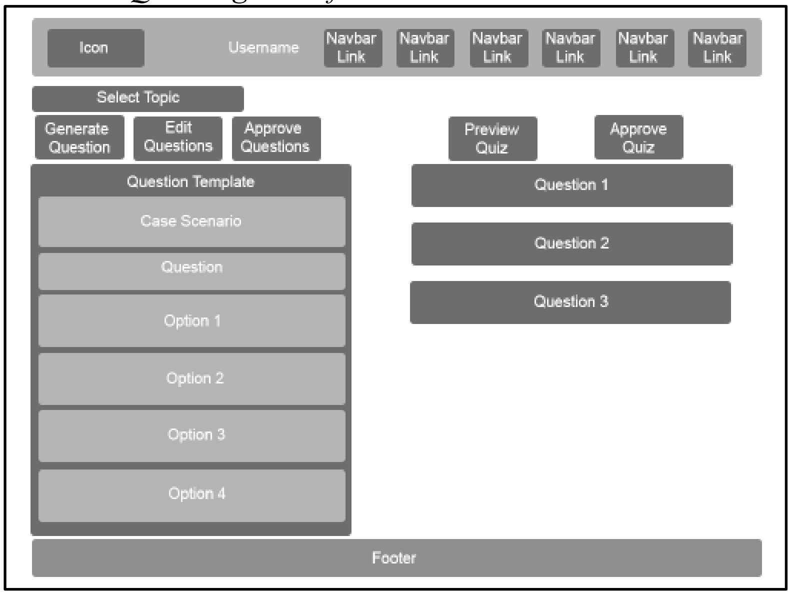
The “Complete Quiz” page design, as portrayed in Figure 11, was also designed during the fifth sprint and is prepared to display questions of the quiz, one at a time, during the quiz-taking process. Each question presents a case scenario, a question, and multiple-choice options with corresponding radio buttons for selection. Additionally, to allow for intuitive navigation, there are buttons to move to the next question or return to the previous one. A progress indicator informs the user of their current position within the sequence of the questions. When reaching the last question, a submission button prompts the user to submit their answers. Then, the final view, instead of another question, presents the user's score, a pass/fail message, and detailed feedback.

Figure 11
Complete Quiz Page Wireframe



The 'Generate Quiz' page, as illustrated in Figure 12, was designed in Sprint 3 and is a dual-aspect layout where the left segment is dedicated to question generation. This area displays a newly generated quiz question, including case scenarios, possible answers, and feedback for each choice, with indicators marking the correct answers. There are buttons to allow trainers to generate, edit, and approve the question. On the right segment, a list accumulates approved questions, and there is a button to approve the quiz and send it to the quiz repository.

Figure 12
Generate Quiz Page Wireframe



The 'Quiz Repository' page, as presented in Figure 13, was designed during sprint 4 and is intended for trainers to browse the generated quizzes to be assigned to chefs. Each quiz entry has buttons to view the quiz's details, assign it to chefs, and track its history. Assigning a quiz opens the assigning quiz template, as shown in Figure 13, where trainers can set parameters like deadline and passing criteria and select the chef or group of chefs to assign the quiz.

Figure 13
Quiz Repository Page Wireframe

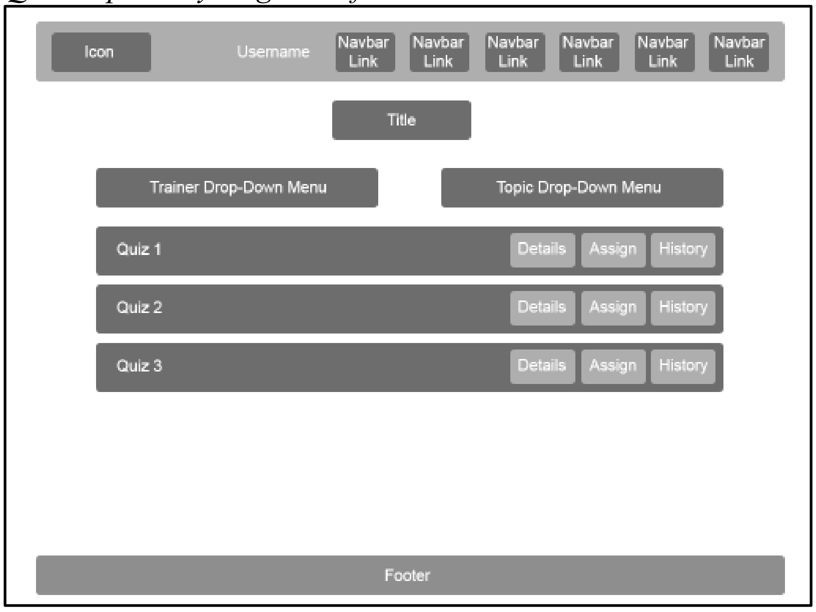


Figure 14
Assign Quiz Template Wireframe

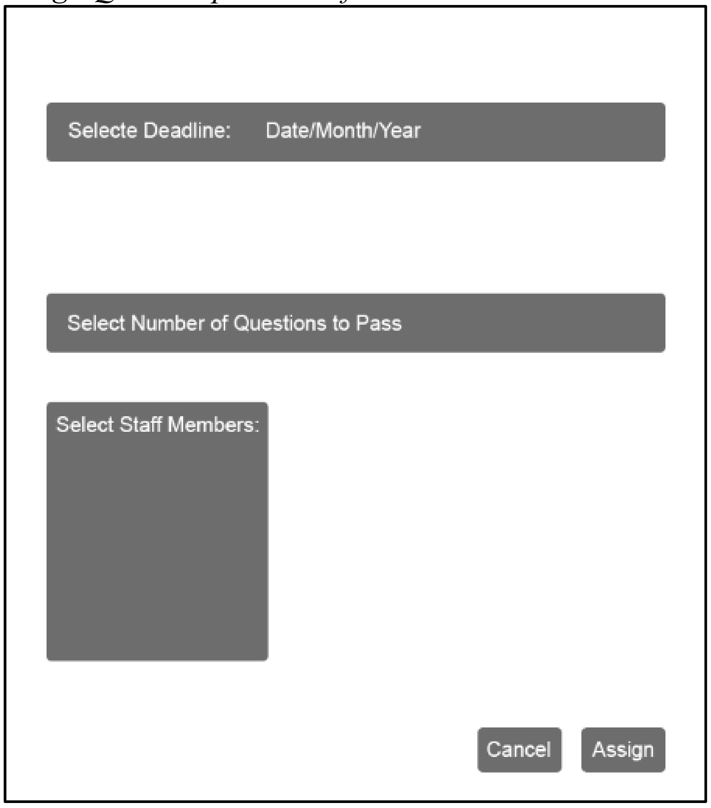
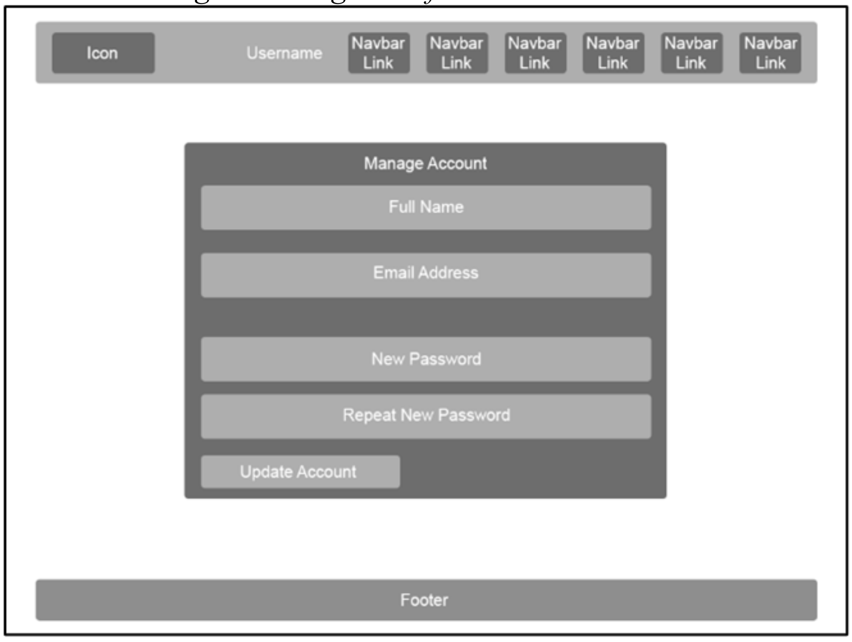

Selecte Deadline: Date/Month/Year
Select Number of Questions to Pass
Select Staff Members:
Cancel
Assign

Figure 15 demonstrates the account management page design, created at the start of the second sprint. It offers a straightforward form where users can update their personal details, such as name, email, and password. This allows users to maintain their profiles with minimal effort and ensures a user-friendly experience.

Figure 15
Account Management Page Wireframe

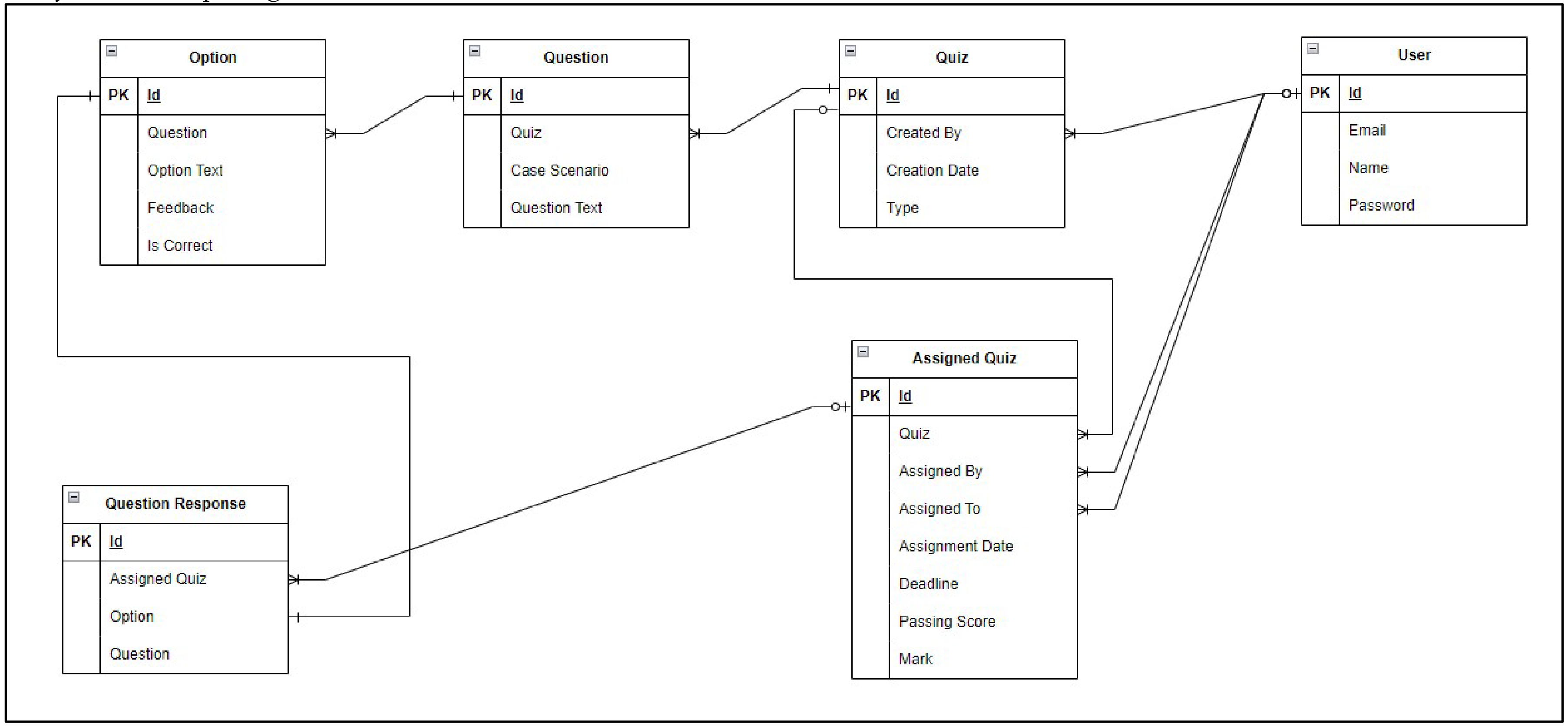


3.3. Database Design

The Entity-Relationship Diagram (ERD) in Figure 16 outlines the design of the application database, where users can create, assign, and take quizzes. This design organises quizzes into a hierarchy of questions and options. 'Assigned Quizzes' are individual instances of a quiz assigned to a user, allowing the reusability of quizzes. Additionally, 'Question Responses' track each user's selected option for each question of an assigned quiz.

The database design sequentially adheres to the principles of normalisation up to the 3NF (Third Normal Form). The 1NF (Fist Normal Form) is achieved by ensuring each table has a primary key and implementing data atomicity. 2NF (Second Normal Form) ensures any non-primary key attribute depends fully on the primary key, avoiding partial dependencies. Lastly, 3NF (Third Normal Form) is achieved by ensuring each non-primary key attribute depends directly on the primary key, avoiding transitive dependencies (Elmasri & Navathe, 2017).

Figure 16
Entity-Relationship Diagram



Chapter 4. Development and Implementation

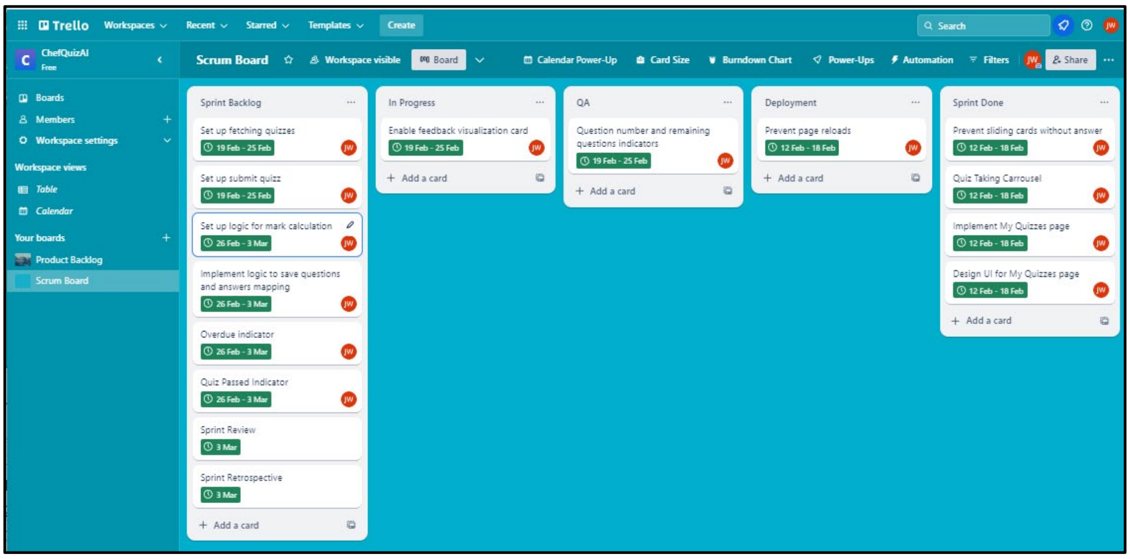
The application's design was implemented using a suite of development tools, each selected for its specific capabilities and compatibility with the project's requirements. The core of the application was developed using Symfony, a PHP framework. This choice was motivated by familiarity with the tool and Symfony's robust ecosystem and extensive documentation. Additionally, Symfony enforcement of the MVC (Model-View-Controller) pattern was crucial for separating the concerns of the application's logic, interface, and control flow, enhancing maintainability and scalability over time (Gamma, Helm, Johnson, & Vlissides, 1995).

In the application's backend, data management and storage were handled by MySQL, a widely used relational database management system. MySQL was selected for its familiarity, reliability, and performance, providing the necessary mechanisms for queries, ensuring data can be retrieved efficiently, and data integrity, ensuring the application's data is stored securely. The OpenAI GPT-4 API was chosen to provide the application with quiz content generation capabilities. The choice to use GPT-4's capabilities was motivated by the findings during the literature review regarding its generative NLP capability advantages over other options and the desire to incorporate cutting-edge technology that could provide real-time, original, accurate and precise content to the quizzes. The API offers several endpoints, but the Chat Completions endpoint was chosen for its out-of-the-box optimisation for conversational tasks, making it ideal for generating interactive, context-aware quiz content (https://platform.openai.com/docs/api-reference/chat/create)

On the front end, the application's user interface was created using Twig templates due to the integration within Symfony and styled with Bootstrap, a front-end framework known for its responsive design templates. Bootstrap was chosen to expedite the development process while ensuring the application is accessible and provides a consistent user experience across various devices and screen sizes. Moreover, client-side interactivity was enhanced with JavaScript to create dynamic elements and interactions on the application's front end, ensuring an engaging user experience.

Regarding the execution of Agile, the chosen software development methodology, Trello, a project management tool, was used to simulate a Scrum board environment and provide a visual and flexible way to manage tasks. As shown in Figure 17, the board was organised into several lists, each corresponding to different stages of the Scrum workflow. At the start of each sprint, selected items from a Product Backlog were added to the Sprint Backlog. As work on a task commenced, they were moved to the In Progress list, indicating that development was underway. Upon completion, tasks were moved to the QA (Quality Assurance) list for testing and quality checks to ensure each feature met the requirements. Successfully tested tasks were then moved to the Deployment list, marking them ready for release into the production environment. Each sprint also had actionable items for Sprint Retrospective and Sprint Review. Any feedback that required further attention was sent back to the Product Backlog.

Figure 17
Scrum Board



Sprint Backlog
- Set up fetching quizzes — 19 Feb - 25 Feb
- Set up submit quizz — 19 Feb - 25 Feb
- Set up logic for mark calculation — 26 Feb - 3 Mar
- Implement logic to save questions and answers mapping — 26 Feb - 3 Mar
- Overdue indicator — 26 Feb - 3 Mar
- Quiz Passed Indicator — 26 Feb - 3 Mar
- Sprint Review — 3 Mar
- Sprint Retrospective — 3 Mar

In Progress
- Enable feedback visualization card — 19 Feb - 25 Feb

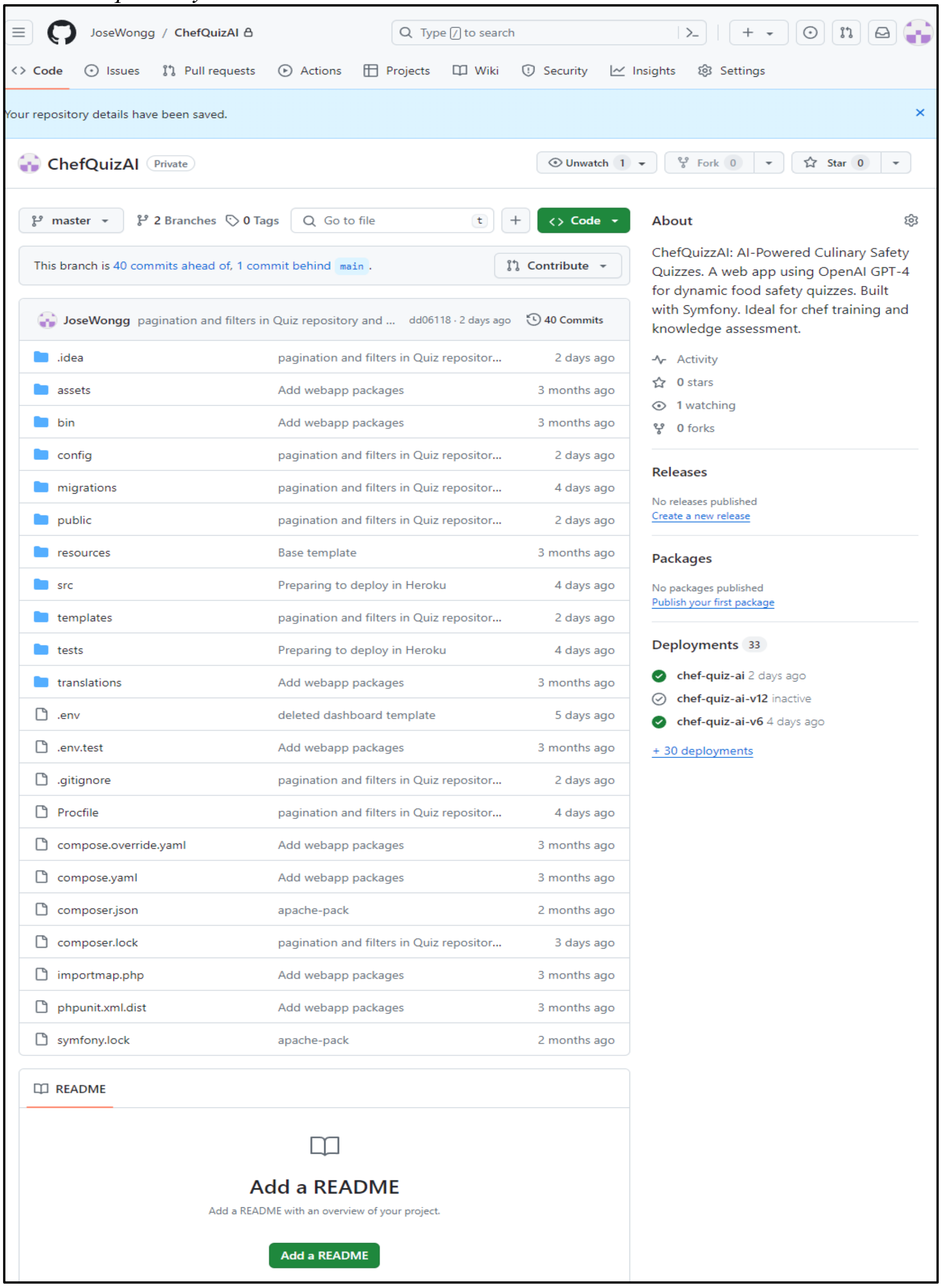
QA
- Question number and remaining questions indicators — 19 Feb - 25 Feb

Deployment
- Prevent page reloads — 12 Feb - 18 Feb

Sprint Done
- Prevent sliding cards without answer — 12 Feb - 18 Feb
- Quiz Taking Carrousel — 12 Feb - 18 Feb
- Implement My Quizzes page — 12 Feb - 18 Feb
- Design UI for My Quizzes page — 12 Feb - 18 Feb

Using GitHub for version control, as shown in Figure 18, allowed for tracking changes in the code base over time. GitHub was chosen due to its familiarity, popularity, and possibility of integrating with Heroku, the chosen deployment platform, for establishing an automatic deployment pipeline. Heroku was selected for its simplicity and efficiency in managing cloud-based applications, facilitating continuous deployment and integration, and allowing for frequent updates and easy scalability in the future as the user base grows.

Figure 18
GitHub Repository



JoseWongg / ChefQuizAI

Your repository details have been saved.

ChefQuizAI Private

master · 2 Branches · 0 Tags

This branch is 40 commits ahead of, 1 commit behind `main`.

JoseWongg pagination and filters in Quiz repository and ... dd06118 · 2 days ago · 40 Commits

.idea	pagination and filters in Quiz repositor...	2 days ago
assets	Add webapp packages	3 months ago
bin	Add webapp packages	3 months ago
config	pagination and filters in Quiz repositor...	2 days ago
migrations	pagination and filters in Quiz repositor...	4 days ago
public	pagination and filters in Quiz repositor...	2 days ago
resources	Base template	3 months ago
src	Preparing to deploy in Heroku	4 days ago
templates	pagination and filters in Quiz repositor...	2 days ago
tests	Preparing to deploy in Heroku	4 days ago
translations	Add webapp packages	3 months ago
.env	deleted dashboard template	5 days ago
.env.test	Add webapp packages	3 months ago
.gitignore	pagination and filters in Quiz repositor...	2 days ago
Procfile	pagination and filters in Quiz repositor...	4 days ago
compose.override.yaml	Add webapp packages	3 months ago
compose.yaml	Add webapp packages	3 months ago
composer.json	apache-pack	2 months ago
composer.lock	pagination and filters in Quiz repositor...	3 days ago
importmap.php	Add webapp packages	3 months ago
phpunit.xml.dist	Add webapp packages	3 months ago
symfony.lock	apache-pack	2 months ago

README

Add a README

Add a README with an overview of your project.

About

ChefQuizzAI: AI-Powered Culinary Safety Quizzes. A web app using OpenAI GPT-4 for dynamic food safety quizzes. Built with Symfony. Ideal for chef training and knowledge assessment.

Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Deployments 33

chef-quiz-ai 2 days ago
chef-quiz-ai-v12 inactive
chef-quiz-ai-v6 4 days ago
+ 30 deployments

4.1. Sprint 1: Database, Homepage and Deployment Pipeline

During the first sprint of the Food Safety Training Application's development, the focus was setting up a database for data storage, creating a landing page and navigation bar, and establishing an automated deployment pipeline to ensure the application remains up-to-date and operational.

Implementing these functionalities responds to the user stories portrayed in Table 3.

Table 3
Sprint 1 User Stories

Number	Title	User Type	I want to...	So that…
US1	Web Access	All users	Access the application through a webpage from any device with internet access.	I can use the service without downloading or installing any software.
US2	Homepage	All users	Have the homepage immediately present clear options and information.	I can quickly understand what the application offers.
US3	Application Updates	All users	Have the application consistently available and current, with regular updates.	I always have access to the latest software version without requiring manual updates.
US4	Navbar	All users	Navigate to the sections of the application.	I can easily access the resources that I need.
US5	Data Accuracy	All users	Access relevant and accurate data.	I can trust the information and make informed decisions when interacting with the website.

Consequently, these user stories were translated into usability requirements, as shown in Table 4.

Table 4
Sprint 1 Usability Requirements

Number	Requirement	Description	User Story
UR1	Responsive Web Access	The user should be able to access the application on various devices, adapting its screen size.	US1
UR2	Intuitive Homepage	The homepage should have a minimalistic design that is appealing but not overcrowded.	US2
UR3	Application Updates	The user should not need to manually update the application to access the latest content and version.	US3
UR4	Intuitive Navbar	The user should be able to easily navigate accessing the required pages.	US4
UR5	Interaction Feedback	The user should receive clear guidance when they make an error.	US2, US4

The list of user requirements was then used to elaborate a list of the functionalities the system must provide, as contained in Table 5.

Table 5
Sprint 1 Functional Requirements

Number	Requirement	Description	User Requirement
FR1	Access Application	The system must allow the user to access an entry point through a homepage without downloading or installing software.	UR2
FR2	Navigate Between Sections	The system must allow the user to navigate from one page to another.	UR4
FR3	View Responsive Content	The system must allow users to view content responsively in different screen sizes.	UR1
FR4	Access Accurate and Secured Data	The system must allow the user to access accurate data and guarantee its integrity.	UR5
FR5	Automatic Updates	The system must ensure the user can access the latest version without intervention.	UR3

4.1.1. Database Implementation

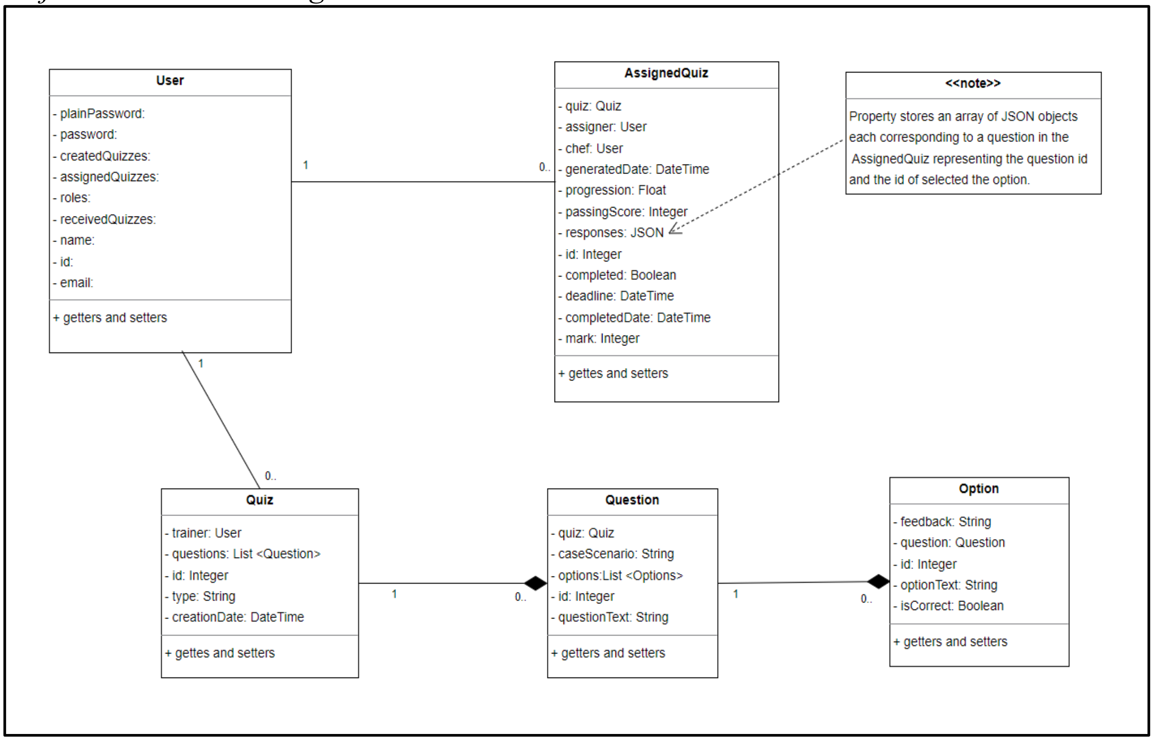
To implement the conceptual database design depicted in Chapter 4's Figure 16, it was converted into an Object-Oriented Symfony-based Object-Relational Mapping (ORM) using Doctrine ORM. This PHP tool allows interaction between object-oriented code and relational databases. This approach allowed a straightforward connection to the database without implementing a dedicated class using the singleton design pattern as it was originally planned.

The process involved the development of entity classes. The User Entity represents users with capabilities to create, assign, and undertake quizzes. The Quiz entity encapsulates the quiz structure with a link to the creating user (trainer) and a collection of Question entities. This setup ensures that the Quiz entity's data integrity is preserved, with each quiz linked to its creator and composed of multiple questions. The Question entity contains a back-reference to its Quiz entity and hosts a collection of Option entities. Similarly, the Option Entity contains a link to a specific Question and represents one possible answer to the question. A pragmatic approach was adopted to represent the Question Responses Table envisaged during the initial design stage as a JSON column of type array within the AssignedQuiz entity, which allowed the mapping of the responses for each question of an assigned quiz at the same time as reducing the application's logic complexity.

During the development of the entity classes, Doctrine's attributes were extensively used to define entity relationships and constraints, ensuring that the underlying database schema was automatically generated using Doctrine migrations to reflect the application's object model, portrayed in Figure 19, simplifying the development process by allowing Symfony commands to create the database schema based on the entity definitions. Using Doctrine ORM also facilitated the implementation of queries without extensive SQL and the use of straightforward validation constraints later during the development process. The schema was deployed on a cloud-hosted MySQL database provided by jawsDB, a PaaS (Platform as a Service), which offered the added advantages of smooth integration with Heroku, the chosen deployment platform, and flexible scalability.

Figure 19

Object Model UML Diagram

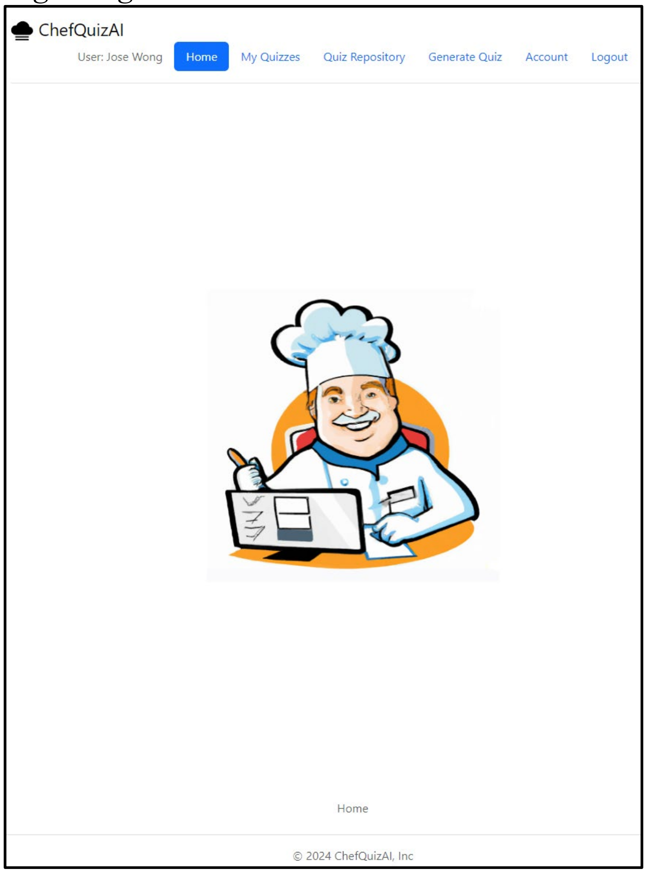


4.1.2. Homepage Implementation

The application's homepage and navigation bar were developed utilising Symfony and Twig, its template engine. As portrayed in Figure 20, the focus was, in line with the design specification, on a simple interface to avoid cognitive overloading and to provide a modern but formal appearance an educational application requires. The homepage prominently displays the application's logo, cantered and optimised for quick loading from the public assets' directory and the navigation bar dynamically adjusts the navigation links based on user authentication privileges, working in tandem with Symfony's Role Based Access Control mechanisms, presenting tailored options like "Home" for unauthenticated users, "My Quizzes" for authenticated users and "Generate Quiz" for trainer users, ensuring a personalised experience through Twig's templates conditional rendering.

The use of Bootstrap ensures that the design is responsive across devices. Template inheritance is utilised to create a consistent and maintainable codebase. This approach allows the base template to define common elements such as the navigation bar and footer, ensuring a uniform look and feel across the application while enabling individual pages, like the homepage, to inherit the base template and insert specific content into predefined blocks for additional content. This method simplifies updates to the site's layout, as changes in the base template automatically propagate to all inheriting templates, enhancing development efficiency and reducing code redundancy.

Figure 20
Login Page



During the implementation and across the remaining sprints, functional testing of the Navbar was carried out to verify that all elements on the homepage and navigation bar loaded correctly and remained clickable, and links, buttons, and dynamic content changed based on user authentication status. Similarly, the homepage and navbar's responsiveness, design, and functionality were tested across different devices and browsers.

4.1.3. Deployment Pipeline Set Up

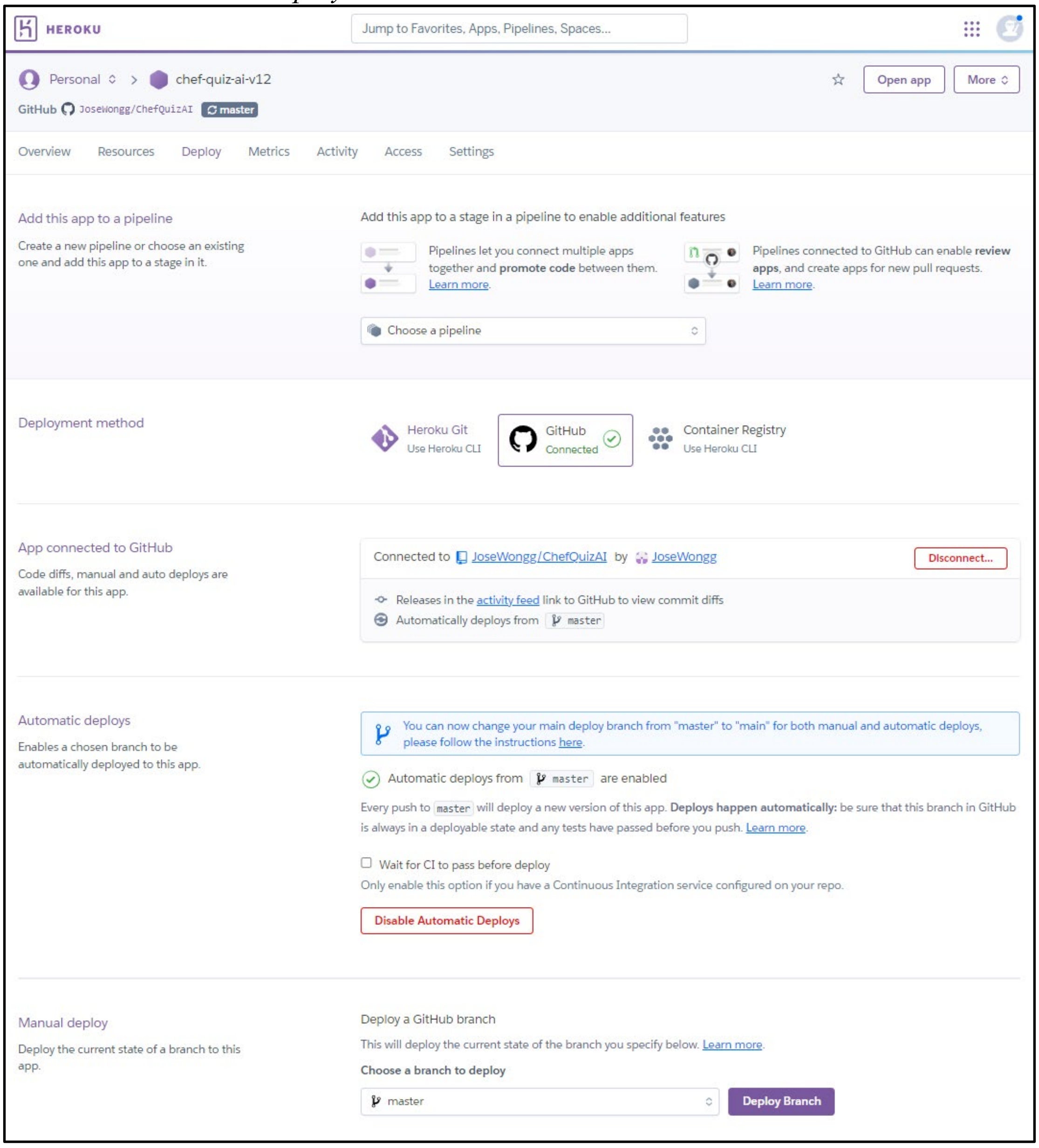
This was probably one of the most challenging aspects of the application development. Initial attempts to deploy the application on a shared physical server resulted in images not rendering and broken navigation links largely due to the lack of server documentation and incompatibility with the file system structure used by Symfony.

A deployment solution was attempted using a Docker container, which, despite having achieved the generation of a Docker image of the application and attempted its deployment several times, also ended up being a failed deployment process.

The solution successfully implemented in the end was using Heroku, one of the most popular PaaS (Platform as a Service) cloud website deployment providers. This solution came with the added benefits of CI/CD (Continuous Integration/Continuous Delivery), as the deployment pipeline allowed the activation of an automatic deployment feature that implies that any changes in the local development environment that are pushed to the GitHub repository are immediately reflected in the production environment as presented in Figure 21.

Additionally, the Heroku dashboard allowed access to production environment logs, activity and performance statistics and a straightforward set-up of production environment variables such as database and API keys and the path to the SSL certificate used by the application. Other potential benefits of using Heroku include flexibility, scalability, and integration of automated tests into the deployment pipeline. A thorough testing was carried out after the deployment pipeline setup and after each sprint, ensuring the rendering of static assets, functioning routes and integrity of the application functionalities.

Figure 21
Heroku's Continuous Deployment Dashboard



Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Deployment method

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to JoseWongg/ChefQuizAI by JoseWongg

Releases in the activity feed link to GitHub to view commit diffs

Automatically deploys from `master`

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

Automatic deploys from `master` are enabled

Every push to `master` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. Learn more.

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

4.2. Sprint 2: User Management

In the second sprint of our Food Safety Training Application development, we focused on implementing essential user management functionalities, including authentication (login and logout), registration, and account management, to respond to the user stories in Table 6.

Table 6
Sprint 2 User Stories

Number	Title	User Type	I want to…	So that…
US1	Login	All Users	Log in through a webpage using email and password.	I can access the relevant functionalities and data in the application.
US2	Login and Registration Error	All Users	Receive feedback.	I know what went wrong.
US3	Logout	Authenticated Users	Log out.	My session ends.
US4	Registration	New Users	Register a new account.	I can interact with the application.
US5	Password Hashing	New Users	Have a secure password.	My data is safe.
US6	Account Management	Authenticated Users	Update my credentials.	My personal information is accurate.

Accordingly, these user stories were converted into a set of usability requirements, as shown in Table 7

Table 7
Sprint 2 Usability Requirements

Number	Requirement	Description	User Story
UR1	Intuitive Login Form	The login page should have a clear layout, placing required fields in a prominent location to allow users to log in.	US1
UR2	Error Message Clarity	Users should be able to see error messages near the input fields.	US2
UR3	Visible Logout	Users should be able to log out from any page, and the session end should be evident.	US3
UR4	Simplified Registration Form	The registration experience should be simple and concise.	US4
UR5	Secure Password Practices	The user should be made aware of password constraints.	US5
UR6	Login, Registration and Account Update Confirmation	Login, Registration and Profile changes should be confirmed with a clear message.	US6

This list of usability requirements was then used to elaborate on the functionalities that the system must provide, as contained in Table 8.

Table 8
Sprint 2 Functional Requirements

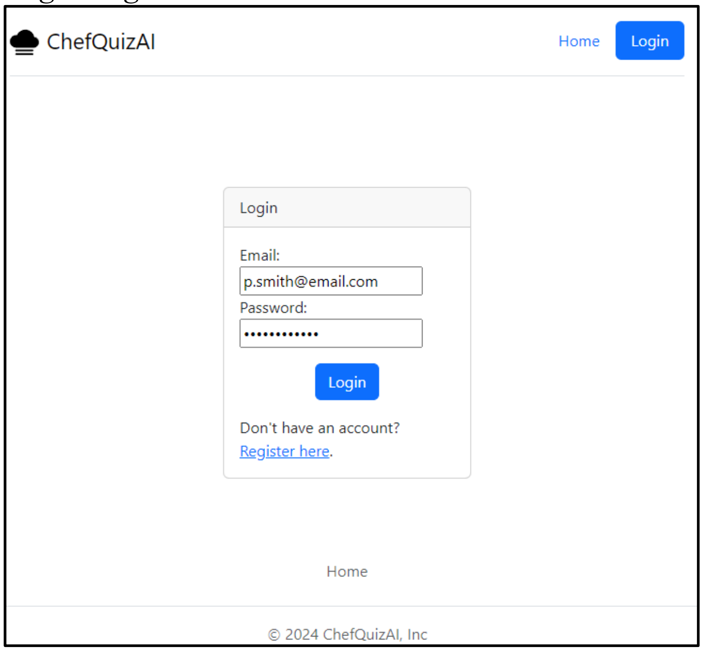
Number	Requirement	Description	User Requirement
FR1	User Authentication	The system must verify login credentials against the database and provide access according to the user-predefined roles.	UR1
FR2	Display Login, Registration, and Account Update Errors	The system must display appropriate errors when users enter wrong input during registration, account update and log-in.	UR2
FR3	Logout Functionality	The system must allow users to terminate sessions, clear session data, and redirect users to the homepage.	UR3
FR4	User Registration	The system must provide functionality to register new users and store their profiles in the database.	UR4
FR5	Password Security	The system must hash new passwords and store them securely.	US5
FR6	Account Update	The system must allow users to update their profile details and safely store changes.	UR6
FR7	Login, Registration, and Account Update Confirmation	The system must provide confirmation when the user logs in, registers and changes their profile details.	UR1, US3, US4

4.2.1. Authentication Functionality Implementation

We used Symfony's Security Bundle to manage user authentication. This bundle has built-in mechanisms that allow efficient login and logout functionality setup, securely handling user sessions and access controls. The login page, presented in Figure 22, was created with a Twig template, inheriting from the base layout. This page provides a form for users to enter their email and password. It also displays any authentication errors and includes a link to the registration page for new users. A Form Type class defines the structure of the login form. It specifies fields for the email and password, validating that inputs are in the correct format. When the user enters incorrect login credentials, error messages are displayed directly on the login form, thanks to the template's integration with Symfony's AuthenticationUtils service.

In line with the MVC pattern, a controller handles the rendering of the login form and processing login/logout requests. The AuthenticationUtils service also allows the application to obtain any last authentication error and the last entered username, improving user experience by pre-populating the username field upon an authentication failure. Symfony's security system handles the logout method in the background without requiring any code implementation other than indicating the page to be rendered after logging out. The Form Type class also implements CSRF (Cross-Site Request Forgery), ensuring secure form submissions by verifying each request's origin and preventing external attacks from exploiting authenticated sessions.

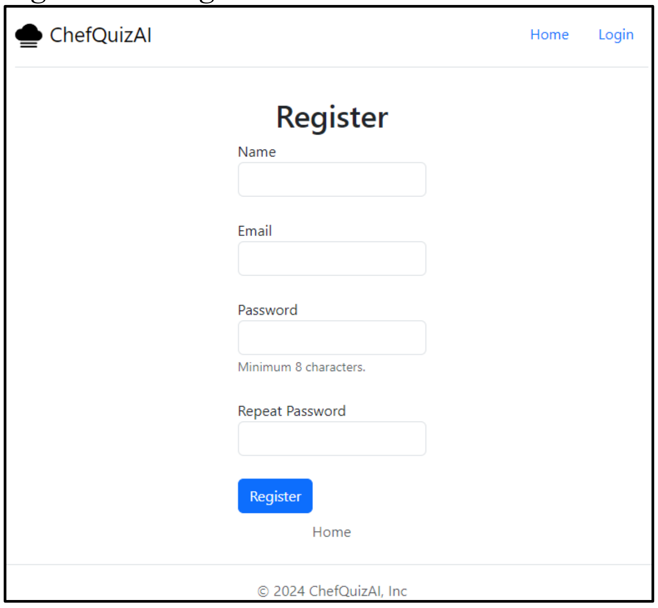
Figure 22
Login Page



4.2.2. Registration Functionality Implementation

The registration functionality allows new users to create accounts using a form displayed on a page created with a Twig template. As shown in Figure 23, the form collects user details such as name, email, and password. Validation feedback is provided directly on this form for any input errors. A Form Type class defines the form's structure and validation rules, ensuring the email is in the correct format and the password meets specified criteria. The repeat password input field confirms the user's password choice. The login form submission is handled by a controller, which uses a custom-implemented password hasher service for secure password handling, ensuring passwords are hashed before being persisted in the database.

Figure 23
Registration Page



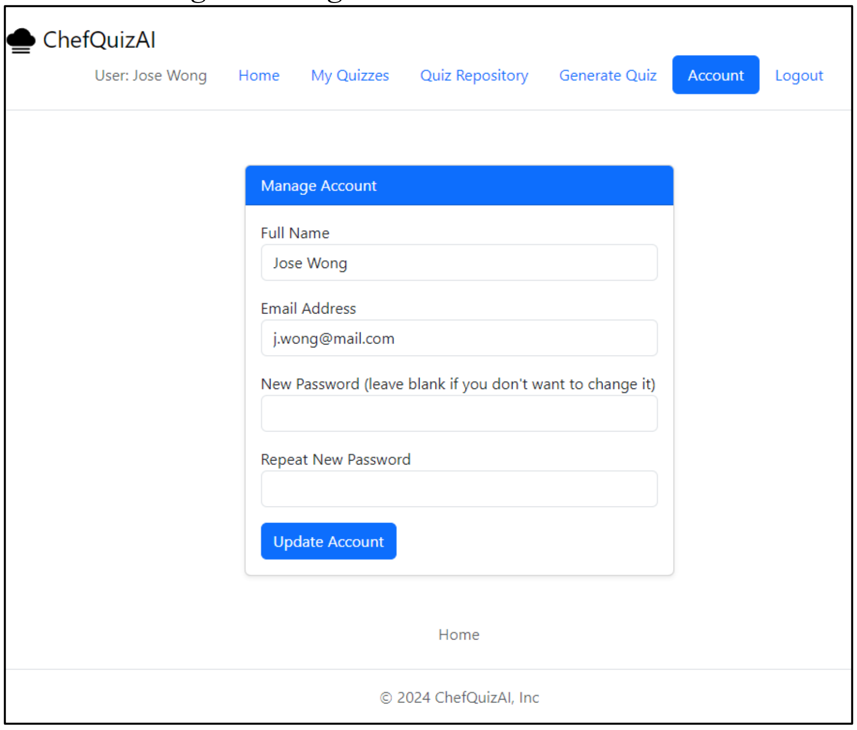
The User entity includes validation constraints for the email and password fields to maintain data integrity. These constraints ensure email uniqueness, proper format, and a minimum password length. Errors from these validations are clearly communicated to the user through the registration form interface. Upon successful registration, users are redirected to the login page. A JavaScript alert informs them of successful account creation, enhancing the user interface by confirming the registration outcome.

4.2.3. Account Management Functionality Implementation

The Account Management functionality was developed to allow users to update their profile details, including their full name, email address, and password. This feature uses a Twig template for the user interface, a Form Type class to define the form structure and validation, and a controller to handle the form submission and data processing.

The Twig template, portrayed in Figure 24, contains a form for users to update their account information. The fields are optional, allowing users to leave them blank if they do not wish to change them. A second input is required to confirm a new password. The controller handles the form submission, loads the current user's data into the form, updates the user entity with the submitted data, and persists in the changes to the database. A success message is displayed upon successful update, and the user is redirected to the profile edit page.

Figure 24
Account Management Page



Manage Account

Full Name
Jose Wong

Email Address
j.wong@mail.com

New Password (leave blank if you don't want to change it)

Repeat New Password

Update Account

A particularly challenging aspect of this implementation was ensuring that when users update their passwords, the controller automatically re-authenticates them to ensure the session remains instead of needing them to log in again. We used Symfony's TokenStorageInterface to update the login session with the new password to achieve this.

4.2.4. Role-Based Access Control

Role-Based Access Control (RBAC) was implemented in the application to ensure users have access only to the features and data relevant to their roles. This was achieved by defining access control routes in the project's security configuration file, a core feature of Symfony's security mechanism. Publicly accessible routes like /login, /register, and homepage are available to all, including unauthenticated users. Routes for creating, saving, and managing quizzes are restricted to users with the ROLE_TRAINER role. To facilitate this functionality, the User entity was modified to add a new property holding an Array of roles. Consequently, this change was migrated to the database using Doctrine ORM attributes to create a new column for the users' roles.

ROLE_TRAINER is typically assigned to users like head chefs and kitchen or restaurant managers who can create and assign quizzes. The decision was taken to implement the assignment of this role in a way that the role is granted by the registration controller when a new user creates a new account using the @chefquizai.trainer.com email domain, which presumably can be assigned by the client to authorised users. The application restricts all other routes to authenticated users, ensuring that sensitive features and data are protected and only accessible to users who have logged in.

4.3. Sprint 3: Generate Quiz Functionality

During this Sprint, the effort focused on the critical Generate Quiz functionality, which addressed all the user stories identified in Table 9.

Table 9 *Sprint 3 User Stories*

Number	Title	User Type	I want to…	So that…
US1	Generate Question	Trainer	Automatically generate a quiz question.	I can offer engaging and accurate assessment resources.
US2	Edit Question	Trainer	Edit a generated question.	The question is accurate and relevant.
US3	Approve Question	Trainer	Approve a question before it is added to a quiz.	I ensure the question meets the required standards before adding it to the quiz.
US4	Delete Question	Trainer	Delete a generated question.	I can avoid questions that do not meet the required standards being used for the quiz.
US5	Preview Quiz	Trainer	Preview the entire quiz before final approval.	I ensure all the questions are appropriate and accurate.
US6	Generate Quiz	Trainer	Compile a quiz.	I can use it to assess food safety knowledge and promote a culture of food safety in my team.

Subsequently, these user stories were used to produce a set of usability requirements, as shown in Table 10.

Table 10
Sprint 3 Usability Requirements

Number	Requirement	Description	User Story
UR1	Intuitive Question Generation	The process of selecting a topic and generating a question should be straightforward.	US1
UR2	Straightforward Question Editing	Updating a question should be an intuitive and easy process.	US2
UR3	Easy Question Approval	Trainers should be able to view and add a question to the quiz easily.	US3
UR4	Straightforward Question Deleting	Trainers should be able to delete a question easily.	US4
UR5	Comprehensive Quiz Previewing	Trainers should be able to visualise the entire quiz after adding each question.	US5
UR6	Streamline Quiz Generation	The process of compiling a quiz should be intuitive and straightforward.	US6

Afterwards, the usability requirements were turned into a list of the functionalities that the system must provide, as contained in Table 11

Table 11
Sprint 3 Functional Requirements

Number	Requirement	Description	User Requirement
FR1	Question Generation	The system must provide a service to generate questions automatically based on selected topics.	UR1
FR2	Question Editing	The system must implement functionality to edit generated questions.	UR2
FR3	Question Approval	The system must provide an area to visualise a question and a button to approve it.	UR3
FR4	Question Deletion	The system must provide a button to delete a question.	UR4
FR5	Quiz Preview	The system must provide the functionality to preview a quiz.	UR5
FR6	Quiz Generation	The system must provide a dropdown list to select a topic and a button to generate a question.	UR6

The challenge was providing the application with the ability to generate original, accurate and engaging quizzes related to different aspects of food safety. The implemented solution involved a front-end Twig template for the trainers to interact with the application and generate quizzes, a JavaScript script to provide the template with event-driven capabilities, a service to connect the application with the OpenAI GPT-4 model AI for generative language capabilities and a controller to serve as a bridge between the template, the service, the database and the involved entities such as Quiz, Question and Option.

4.3.1. Generate Quiz Template Implementation

The template, as demonstrated in Figure 25, presents, on the left side, a section to generate questions. A dropdown list allows the user to select from five topics: Cross-contamination, Cooking Food, Chilling Food, Allergens and Cleaning. After selecting a topic, a button to generate questions triggers the dynamic content generation by fetching a new question based on the selected topic. Once a topic is selected for the first question, the remaining questions must be on the same topic. Additionally, there are buttons to edit and approve questions.

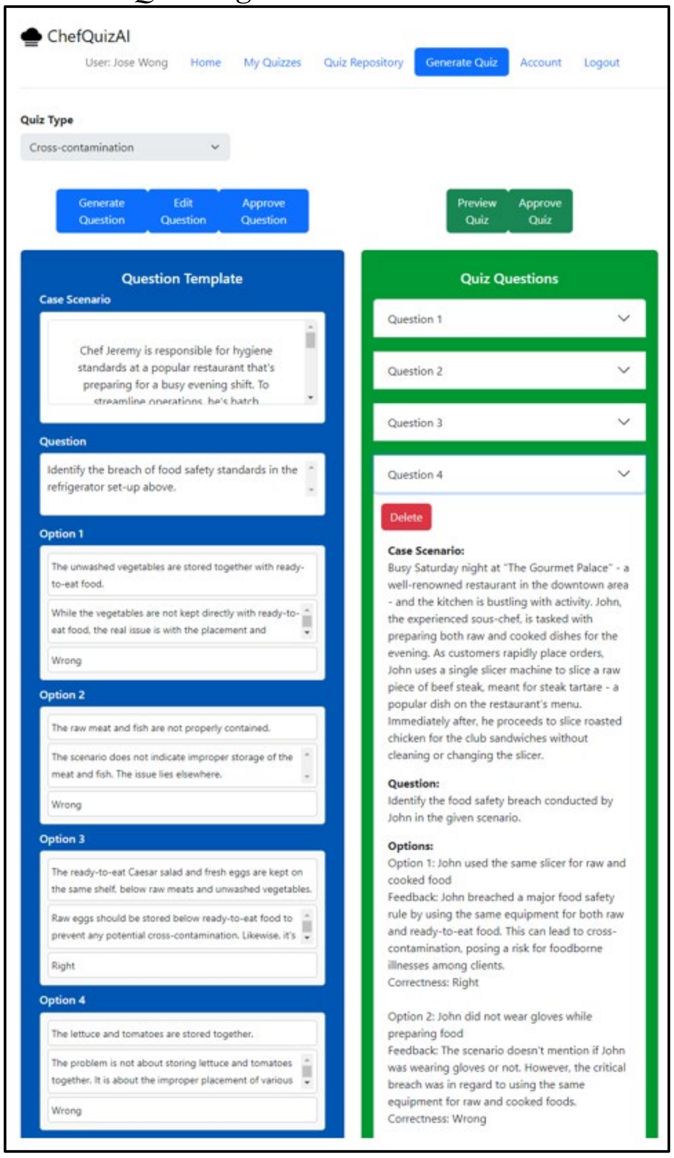
The generated questions include a question scenario, four possible answers, one right and three wrong, and specific feedback for each answer option. Once questions are approved, they are appended to an accordion-style list on the right side of the template, where they can be displayed or hidden individually.

Once the desired number of questions are generated, the quiz can be approved and persisted and is ready to be assigned from the Quiz Repository page. To respond to received feedback, functionality to preview a complete quiz before approval and to delete questions already added to the accordion were implemented. The preview quiz functionality was implemented using another Twig template embedded into a Bootstrap modal, which displays a pop-up window showing all the questions of the quiz, as demonstrated in Figure 26.

A particularly challenging aspect of this implementation was to allow the page to enter into edit mode to allow editing the question fields without refreshing the page and losing all the questions already appended to the quiz. However, editing as well as approving questions, and completing quiz functionalities were essential since the questions are generated by GPT-4, an NLP (Natural Language Processing) model; hence, it was necessary to adopt a human-in-the-loop approach to integrate human expertise and make sure the generated content was accurate (Grønsund & Aanestad, 2020). The solution involved a JavaScript function to set the question fields to editable and checks to avoid inconsistent changes, such as avoiding setting all the options of answer to "incorrect" status.

Figure 25

Generate Quiz Page



Quiz Type
Cross-contamination

Case Scenario
Chef Jeremy is responsible for hygiene standards at a popular restaurant that's preparing for a busy evening shift. To streamline operations, he's batch

Question
Identify the breach of food safety standards in the refrigerator set-up above.

Option 1
The unwashed vegetables are stored together with ready-to-eat food.
While the vegetables are not kept directly with ready-to-eat food, the real issue is with the placement and
Wrong

Option 2
The raw meat and fish are not properly contained.
The scenario does not indicate improper storage of the meat and fish. The issue lies elsewhere.
Wrong

Option 3
The ready-to-eat Caesar salad and fresh eggs are kept on the same shelf, below raw meats and unwashed vegetables.
Raw eggs should be stored below ready-to-eat food to prevent any potential cross-contamination. Likewise, it's
Right

Option 4
The lettuce and tomatoes are stored together.
The problem is not about storing lettuce and tomatoes together. It is about the improper placement of various
Wrong

Case Scenario:
Busy Saturday night at "The Gourmet Palace" - a well-renowned restaurant in the downtown area - and the kitchen is bustling with activity. John, the experienced sous-chef, is tasked with preparing both raw and cooked dishes for the evening. As customers rapidly place orders, John uses a single slicer machine to slice a raw piece of beef steak, meant for steak tartare - a popular dish on the restaurant's menu. Immediately after, he proceeds to slice roasted chicken for the club sandwiches without cleaning or changing the slicer.

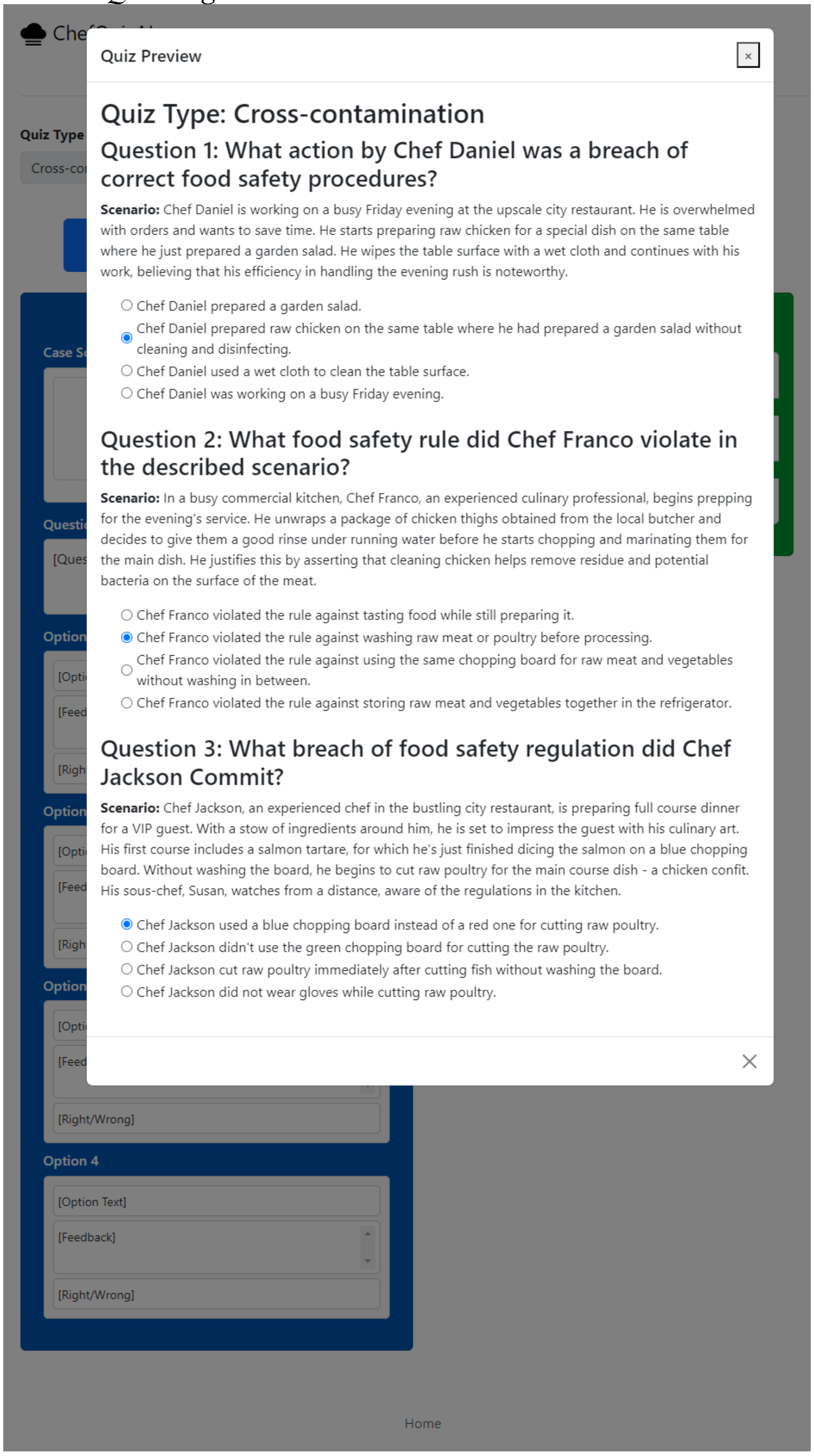
Question:
Identify the food safety breach conducted by John in the given scenario.

Options:
Option 1: John used the same slicer for raw and cooked food
Feedback: John breached a major food safety rule by using the same equipment for both raw and ready-to-eat food. This can lead to cross-contamination, posing a risk for foodborne illnesses among clients.
Correctness: Right

Option 2: John did not wear gloves while preparing food
Feedback: The scenario doesn't mention if John was wearing gloves or not. However, the critical breach was in regard to using the same equipment for raw and cooked foods.
Correctness: Wrong

Figure 26

Preview Quiz Page



Quiz Preview ×

Quiz Type: Cross-contamination

Question 1: What action by Chef Daniel was a breach of correct food safety procedures?

Scenario: Chef Daniel is working on a busy Friday evening at the upscale city restaurant. He is overwhelmed with orders and wants to save time. He starts preparing raw chicken for a special dish on the same table where he just prepared a garden salad. He wipes the table surface with a wet cloth and continues with his work, believing that his efficiency in handling the evening rush is noteworthy.

- ○ Chef Daniel prepared a garden salad.
- ● Chef Daniel prepared raw chicken on the same table where he had prepared a garden salad without cleaning and disinfecting.
- ○ Chef Daniel used a wet cloth to clean the table surface.
- ○ Chef Daniel was working on a busy Friday evening.

Question 2: What food safety rule did Chef Franco violate in the described scenario?

Scenario: In a busy commercial kitchen, Chef Franco, an experienced culinary professional, begins prepping for the evening's service. He unwraps a package of chicken thighs obtained from the local butcher and decides to give them a good rinse under running water before he starts chopping and marinating them for the main dish. He justifies this by asserting that cleaning chicken helps remove residue and potential bacteria on the surface of the meat.

- ○ Chef Franco violated the rule against tasting food while still preparing it.
- ● Chef Franco violated the rule against washing raw meat or poultry before processing.
- ○ Chef Franco violated the rule against using the same chopping board for raw meat and vegetables without washing in between.
- ○ Chef Franco violated the rule against storing raw meat and vegetables together in the refrigerator.

Question 3: What breach of food safety regulation did Chef Jackson Commit?

Scenario: Chef Jackson, an experienced chef in the bustling city restaurant, is preparing full course dinner for a VIP guest. With a stow of ingredients around him, he is set to impress the guest with his culinary art. His first course includes a salmon tartare, for which he's just finished dicing the salmon on a blue chopping board. Without washing the board, he begins to cut raw poultry for the main course dish - a chicken confit. His sous-chef, Susan, watches from a distance, aware of the regulations in the kitchen.

- ● Chef Jackson used a blue chopping board instead of a red one for cutting raw poultry.
- ○ Chef Jackson didn't use the green chopping board for cutting the raw poultry.
- ○ Chef Jackson cut raw poultry immediately after cutting fish without washing the board.
- ○ Chef Jackson did not wear gloves while cutting raw poultry.

4.3.2. Generate Quiz Back End Functionality

The engine behind the question generation functionality was implemented as a service class (Appendix G), which utilises the GuzzleHTTP client library to facilitate making HTTP POST requests to the OpenAI GPT-4 API. Initial testing of the endpoint with PHPUnit proved that the endpoint produced food safety case scenarios and questions in a creative and imaginative style. Although the endpoint offers parameters like temperature, tokens frequency penalty, and presence penalty for fine-tuning, the initial output quality met the project's needs for creative and engaging content out of the box, and the effort later went into making the responses more detailed, using relevant food safety data and prompt engineering techniques.

One of the most challenging aspects of this project was to achieve a response from the GPT-4 model that adhered to specific constraints and requirements regarding structure. The response had to implement a consistent and precise structure, producing a case scenario depicting a breach of food safety best practices, a question, four options of answer, feedback, and an indication of correctness for each option. Attempts to use the functionality embedded into the Chat Completions GPT-4 API endpoint to generate JSON responses did not obtain consistent results. The aim was to find another way to produce a response that could be parsed so that all these elements could be later manipulated separately for display in the template. This required a substantial effort to experiment with the prompt until an acceptable result was achieved by instructing the model to use delimiters like "<<<Case Scenario>>>" and "<<<End Case Scenario>>>” in the response, which made parsing the generated content in the controller easier, allowing the application to extract different components of the quiz question efficiently.

Despite the prompt engineering approach, consisting of including in the prompt the expected structure, achieved well-structured responses most of the time, the testing of the functionality determined that, on average, one in every twenty attempts to generate a question would produce a faulty response structure, sometimes not including the delimiters or perhaps missing chevron characters in the delimiter, causing the parsing process to fail. The solution adopted for this was to implement an error handling mechanism in the controller so that a new attempt to retrieve a response is performed in the background, reducing the incidence of errors reaching the front end from five per cent to 0.25 per cent. If a response that does not conform with the required structure reaches the user, an alert message is displayed to prompt the retrieval of a new question. In addition to structure constraints, the prompt provided instructions to limit the case scenario to below two hundred words, which the language model consistently achieved. This was a workaround as the Chat Completions endpoint allows fine-tuning the max-tokens parameter. However, this constraint applies to the entire response, which would have required making a separate API request just for the case scenario.

Additionally, the need to obtain responses that were related to a food safety topic and appropriate for a quiz was addressed using prompt engineering approaches, providing context in the prompt to specify that the scenario should relate to food safety, and it should include an action in breach of food safety best practices. Additionally, the role assignment prompt engineering pattern was implemented to include a message that defined the AI's role as a helpful assistant for generating detailed case scenarios and actions for a food safety quiz. An additional piece of context was added to the prompt, passing the quiz topic selected by the user on the front end. This avoided implementing the factory design pattern in the code base to generate quizzes for different topics as originally planned. The use of these prompt engineering techniques resulted in a Symfony service that could produce well-shaped and usable quiz questions related to a particular topic. However, upon testing and user interaction, it was clear that the language model consistently generated responses that lacked precise details regarding food safety legal and best practices requirements. Consequently, the functionality was added back to the product backlog during the sprint review for further enhancement at a later sprint.

4.4. Sprint 4: Assign Quiz Functionality

The focus during this Sprint was to respond to the user stories related to trainers assigning quizzes to chefs, as shown in Table 12.

Table 12
Sprint 4 User Stories

Number	Title	User Type	I want to…	So that…
US1	Preview Quiz Before Assignment	Trainer	Preview a generated quiz.	I can ensure the quiz is suitable for the training needs.
US2	Assign Quiz	Trainer	Assign a generated quiz to a chef or group of chefs.	I can assess their food safety knowledge.
US3	Review Assigned Quiz	Trainer	Access the details of a completed quiz.	I can assess the chef's further training needs.

Consequently, these user stories were translated into a set of usability requirements, as shown in Table 13

Table 13
Sprint 4 Usability Requirements

Number	Requirement	Description	User Story
UR1	Preview of Quizzes	The Trainer should be able to view a quiz before assigning it.	US1
UR2	Seamless Quiz Assigning	The chef should be able to easily assign a quiz to the chef(s), setting a passing mark and deadline.	US2
UR3	Easy Access to Past Quizzes	The chef should be able to access any user's past quizzes.	US3

Subsequently, this list of usability requirements was reinterpreted to elaborate a set of functionalities the system must provide, as shown in Table 14.

Table 14

Sprint 4 Functional Requirements

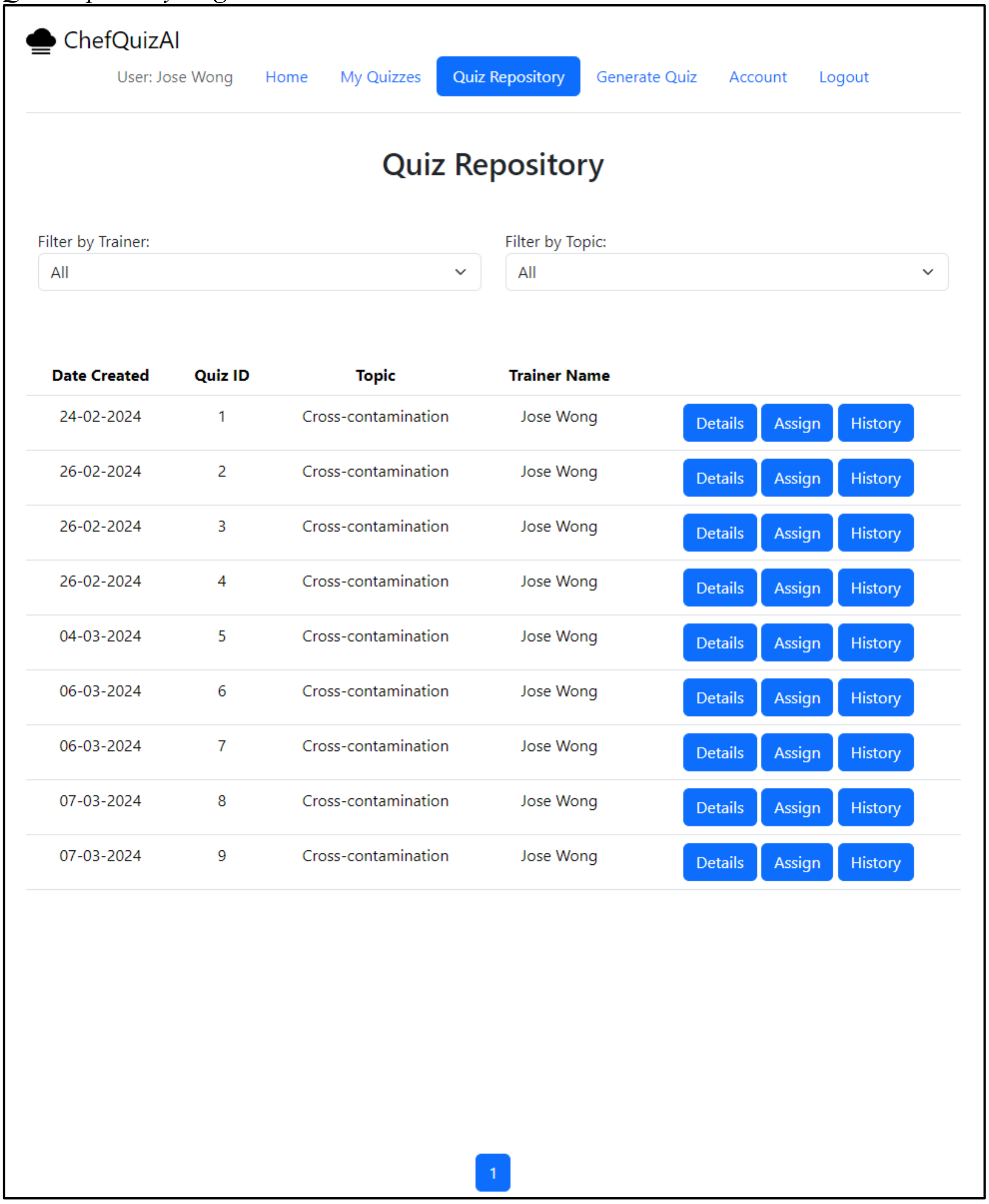
Number	Requirement	Description	User Requirement
FR1	Generated Quiz Display	The system must provide a user interface for interaction and information about a quiz that can be assigned to a chef.	UR1
FR2	Assign Quiz	The system must implement functionality to assign a quiz to a chef or group of chefs, setting a passing mark and deadline.	UR2
FR3	Completed Quizzes Display	The system must implement an interface to provide information about past quizzes already taken by any user.	UR3

This functional requirement comprised the functionalities to list the quiz repository, access a quiz's content, assign a quiz to a chef or group of chefs, visualise a quiz's history, and access an individual instance of an assigned quiz.

As depicted in Figure 27, the quiz repository page lists all the quizzes available to be assigned to a chef. It displays essential information for each quiz, such as the creation date, quiz ID, topic, and the creator's name. A button next to each quiz opens a Bootstrap modal, illustrated in Figure 28, where the trainer can select a deadline, the number of questions to pass, and the name of the chefs to assign the quiz. Upon generating a quiz, a new instance of the generatedQuiz entity, a replica of the Quiz entity, is instantiated for each of the assigned chefs. This approach allowed us to overcome the problem of assigning quizzes while preserving the state of quizzes to be reused in the future.

The Bootstrap modal also implements checks via JavaScript script or the corresponding controller to ensure quizzes are assigned only to chefs who have not taken the same quiz before. Similarly, the selected deadlines can only be a date in the future, and the selected number of questions to pass can be no more than the number of questions in the quiz.

Figure 27
Quiz Repository Page


ChefQuizAI

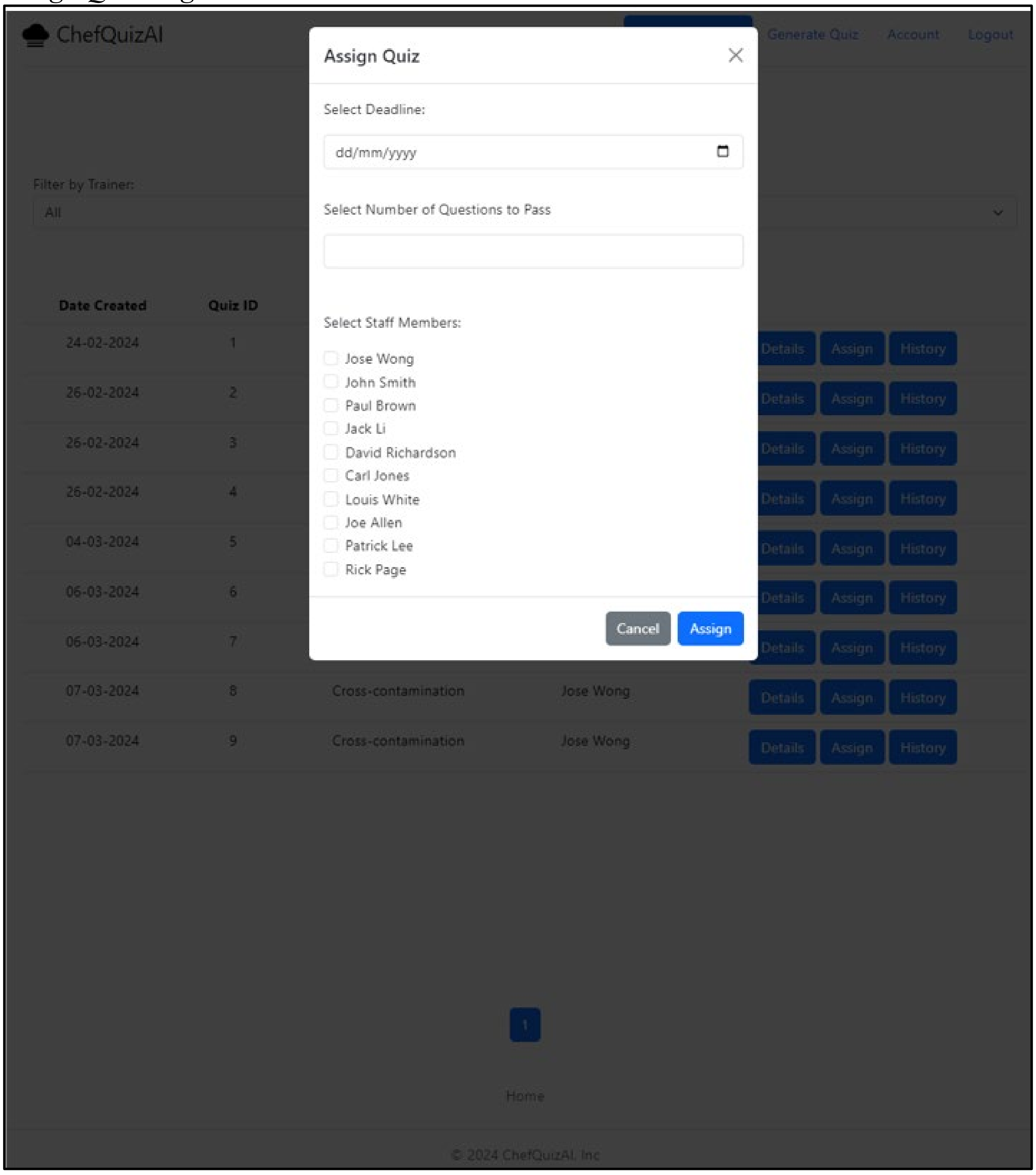
Quiz Repository

Filter by Trainer: All

Filter by Topic: All

Date Created	Quiz ID	Topic	Trainer Name	
24-02-2024	1	Cross-contamination	Jose Wong	Details Assign History
26-02-2024	2	Cross-contamination	Jose Wong	Details Assign History
26-02-2024	3	Cross-contamination	Jose Wong	Details Assign History
26-02-2024	4	Cross-contamination	Jose Wong	Details Assign History
04-03-2024	5	Cross-contamination	Jose Wong	Details Assign History
06-03-2024	6	Cross-contamination	Jose Wong	Details Assign History
06-03-2024	7	Cross-contamination	Jose Wong	Details Assign History
07-03-2024	8	Cross-contamination	Jose Wong	Details Assign History
07-03-2024	9	Cross-contamination	Jose Wong	Details Assign History

Figure 28
Assign Quiz Page



Assign Quiz

Select Deadline:

dd/mm/yyyy

Select Number of Questions to Pass

Select Staff Members:

- Jose Wong
- John Smith
- Paul Brown
- Jack Li
- David Richardson
- Carl Jones
- Louis White
- Joe Allen
- Patrick Lee
- Rick Page

Cancel Assign

Filter by Trainer:

All

Date Created	Quiz ID			
24-02-2024	1			Details Assign History
26-02-2024	2			Details Assign History
26-02-2024	3			Details Assign History
26-02-2024	4			Details Assign History
04-03-2024	5			Details Assign History
06-03-2024	6			Details Assign History
06-03-2024	7			Details Assign History
07-03-2024	8	Cross-contamination	Jose Wong	Details Assign History
07-03-2024	9	Cross-contamination	Jose Wong	Details Assign History

1

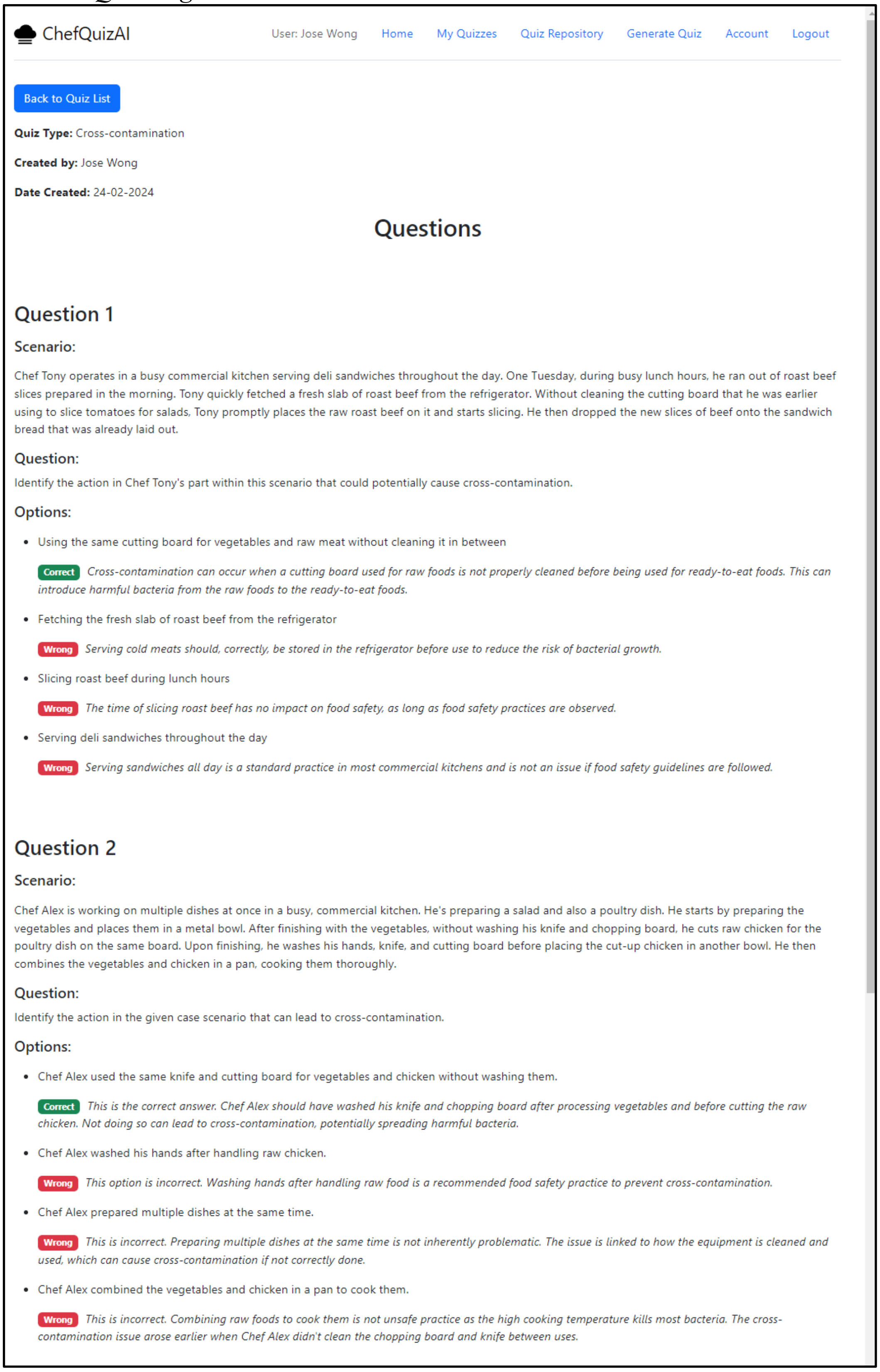
Home

© 2024 ChefQuizAI, Inc

The Quiz repository page also allows the user to open a new Twig template, portrayed in Figure 29, for each quiz listed in the repository, intended to view the content of the quiz before being assigned. A third Twig template, shown in Figure 30, is also accessible from the Quiz repository to view the history of a quiz, listing all the instances where a quiz has been assigned to a chef in the past, including the assignment date, the chef's name, the deadline, the mark achieved, and whether the quiz is overdue or has already been completed. Each of these assigned quizzes can be accessed via a fourth Twig template, which presents the content of an assigned quiz and the options selected by the chef for each question, as presented in Figure 31.

Figure 29

Preview Quiz Page



Quiz Type: Cross-contamination

Created by: Jose Wong

Date Created: 24-02-2024

Questions

Question 1

Scenario:

Chef Tony operates in a busy commercial kitchen serving deli sandwiches throughout the day. One Tuesday, during busy lunch hours, he ran out of roast beef slices prepared in the morning. Tony quickly fetched a fresh slab of roast beef from the refrigerator. Without cleaning the cutting board that he was earlier using to slice tomatoes for salads, Tony promptly places the raw roast beef on it and starts slicing. He then dropped the new slices of beef onto the sandwich bread that was already laid out.

Question:

Identify the action in Chef Tony's part within this scenario that could potentially cause cross-contamination.

Options:

- Using the same cutting board for vegetables and raw meat without cleaning it in between

  Correct *Cross-contamination can occur when a cutting board used for raw foods is not properly cleaned before being used for ready-to-eat foods. This can introduce harmful bacteria from the raw foods to the ready-to-eat foods.*
- Fetching the fresh slab of roast beef from the refrigerator

  Wrong *Serving cold meats should, correctly, be stored in the refrigerator before use to reduce the risk of bacterial growth.*
- Slicing roast beef during lunch hours

  Wrong *The time of slicing roast beef has no impact on food safety, as long as food safety practices are observed.*
- Serving deli sandwiches throughout the day

  Wrong *Serving sandwiches all day is a standard practice in most commercial kitchens and is not an issue if food safety guidelines are followed.*

Question 2

Scenario:

Chef Alex is working on multiple dishes at once in a busy, commercial kitchen. He's preparing a salad and also a poultry dish. He starts by preparing the vegetables and places them in a metal bowl. After finishing with the vegetables, without washing his knife and chopping board, he cuts raw chicken for the poultry dish on the same board. Upon finishing, he washes his hands, knife, and cutting board before placing the cut-up chicken in another bowl. He then combines the vegetables and chicken in a pan, cooking them thoroughly.

Question:

Identify the action in the given case scenario that can lead to cross-contamination.

Options:

- Chef Alex used the same knife and cutting board for vegetables and chicken without washing them.

  Correct *This is the correct answer. Chef Alex should have washed his knife and chopping board after processing vegetables and before cutting the raw chicken. Not doing so can lead to cross-contamination, potentially spreading harmful bacteria.*
- Chef Alex washed his hands after handling raw chicken.

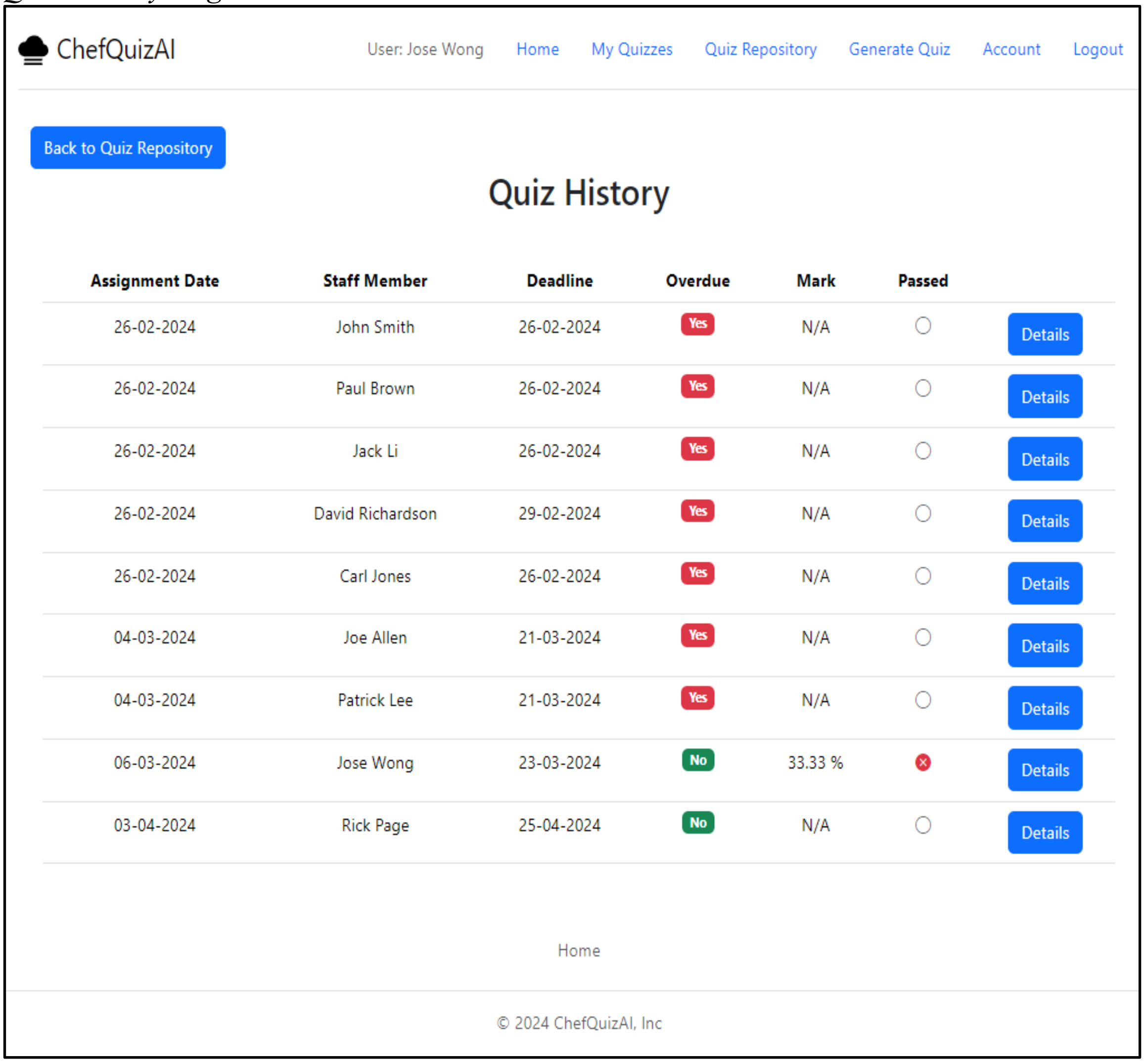
  Wrong *This option is incorrect. Washing hands after handling raw food is a recommended food safety practice to prevent cross-contamination.*
- Chef Alex prepared multiple dishes at the same time.

  Wrong *This is incorrect. Preparing multiple dishes at the same time is not inherently problematic. The issue is linked to how the equipment is cleaned and used, which can cause cross-contamination if not correctly done.*
- Chef Alex combined the vegetables and chicken in a pan to cook them.

  Wrong *This is incorrect. Combining raw foods to cook them is not unsafe practice as the high cooking temperature kills most bacteria. The cross-contamination issue arose earlier when Chef Alex didn't clean the chopping board and knife between uses.*

Figure 30

Quiz History Page



Quiz History

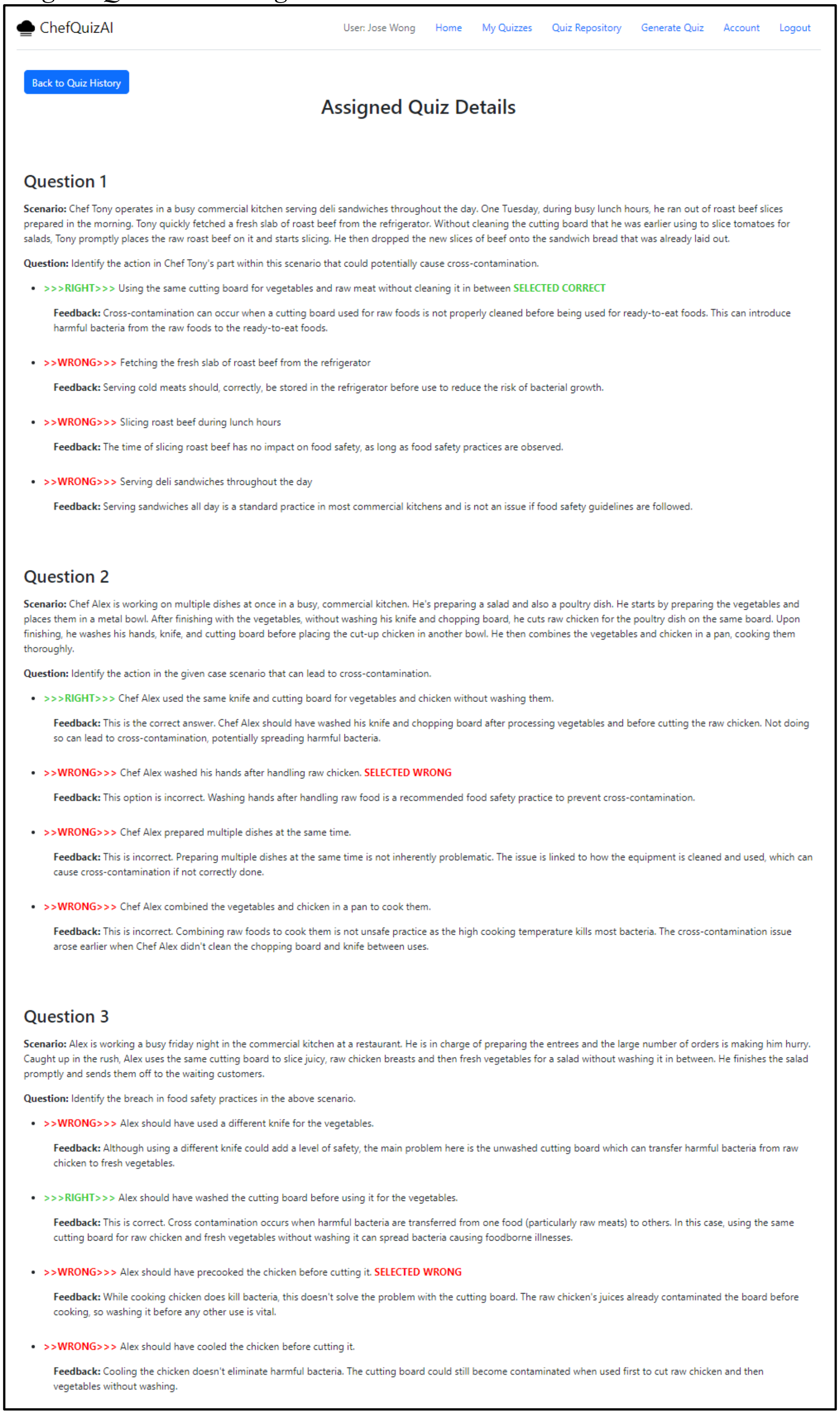
Assignment Date	Staff Member	Deadline	Overdue	Mark	Passed	
26-02-2024	John Smith	26-02-2024	Yes	N/A	○	Details
26-02-2024	Paul Brown	26-02-2024	Yes	N/A	○	Details
26-02-2024	Jack Li	26-02-2024	Yes	N/A	○	Details
26-02-2024	David Richardson	29-02-2024	Yes	N/A	○	Details
26-02-2024	Carl Jones	26-02-2024	Yes	N/A	○	Details
04-03-2024	Joe Allen	21-03-2024	Yes	N/A	○	Details
04-03-2024	Patrick Lee	21-03-2024	Yes	N/A	○	Details
06-03-2024	Jose Wong	23-03-2024	No	33.33 %	✗	Details
03-04-2024	Rick Page	25-04-2024	No	N/A	○	Details

Home

© 2024 ChefQuizAI, Inc

Figure 31

Assigned Quiz Details Page



Assigned Quiz Details

Question 1

Scenario: Chef Tony operates in a busy commercial kitchen serving deli sandwiches throughout the day. One Tuesday, during busy lunch hours, he ran out of roast beef slices prepared in the morning. Tony quickly fetched a fresh slab of roast beef from the refrigerator. Without cleaning the cutting board that he was earlier using to slice tomatoes for salads, Tony promptly places the raw roast beef on it and starts slicing. He then dropped the new slices of beef onto the sandwich bread that was already laid out.

Question: Identify the action in Chef Tony's part within this scenario that could potentially cause cross-contamination.

- >>>RIGHT>>> Using the same cutting board for vegetables and raw meat without cleaning it in between SELECTED CORRECT

  Feedback: Cross-contamination can occur when a cutting board used for raw foods is not properly cleaned before being used for ready-to-eat foods. This can introduce harmful bacteria from the raw foods to the ready-to-eat foods.

- >>WRONG>>> Fetching the fresh slab of roast beef from the refrigerator

  Feedback: Serving cold meats should, correctly, be stored in the refrigerator before use to reduce the risk of bacterial growth.

- >>WRONG>>> Slicing roast beef during lunch hours

  Feedback: The time of slicing roast beef has no impact on food safety, as long as food safety practices are observed.

- >>WRONG>>> Serving deli sandwiches throughout the day

  Feedback: Serving sandwiches all day is a standard practice in most commercial kitchens and is not an issue if food safety guidelines are followed.

Question 2

Scenario: Chef Alex is working on multiple dishes at once in a busy, commercial kitchen. He's preparing a salad and also a poultry dish. He starts by preparing the vegetables and places them in a metal bowl. After finishing with the vegetables, without washing his knife and chopping board, he cuts raw chicken for the poultry dish on the same board. Upon finishing, he washes his hands, knife, and cutting board before placing the cut-up chicken in another bowl. He then combines the vegetables and chicken in a pan, cooking them thoroughly.

Question: Identify the action in the given case scenario that can lead to cross-contamination.

- >>>RIGHT>>> Chef Alex used the same knife and cutting board for vegetables and chicken without washing them.

  Feedback: This is the correct answer. Chef Alex should have washed his knife and chopping board after processing vegetables and before cutting the raw chicken. Not doing so can lead to cross-contamination, potentially spreading harmful bacteria.

- >>WRONG>>> Chef Alex washed his hands after handling raw chicken. SELECTED WRONG

  Feedback: This option is incorrect. Washing hands after handling raw food is a recommended food safety practice to prevent cross-contamination.

- >>WRONG>>> Chef Alex prepared multiple dishes at the same time.

  Feedback: This is incorrect. Preparing multiple dishes at the same time is not inherently problematic. The issue is linked to how the equipment is cleaned and used, which can cause cross-contamination if not correctly done.

- >>WRONG>>> Chef Alex combined the vegetables and chicken in a pan to cook them.

  Feedback: This is incorrect. Combining raw foods to cook them is not unsafe practice as the high cooking temperature kills most bacteria. The cross-contamination issue arose earlier when Chef Alex didn't clean the chopping board and knife between uses.

Question 3

Scenario: Alex is working a busy friday night in the commercial kitchen at a restaurant. He is in charge of preparing the entrees and the large number of orders is making him hurry. Caught up in the rush, Alex uses the same cutting board to slice juicy, raw chicken breasts and then fresh vegetables for a salad without washing it in between. He finishes the salad promptly and sends them off to the waiting customers.

Question: Identify the breach in food safety practices in the above scenario.

- >>WRONG>>> Alex should have used a different knife for the vegetables.

  Feedback: Although using a different knife could add a level of safety, the main problem here is the unwashed cutting board which can transfer harmful bacteria from raw chicken to fresh vegetables.

- >>>RIGHT>>> Alex should have washed the cutting board before using it for the vegetables.

  Feedback: This is correct. Cross contamination occurs when harmful bacteria are transferred from one food (particularly raw meats) to others. In this case, using the same cutting board for raw chicken and fresh vegetables without washing it can spread bacteria causing foodborne illnesses.

- >>WRONG>>> Alex should have precooked the chicken before cutting it. SELECTED WRONG

  Feedback: While cooking chicken does kill bacteria, this doesn't solve the problem with the cutting board. The raw chicken's juices already contaminated the board before cooking, so washing it before any other use is vital.

- >>WRONG>>> Alex should have cooled the chicken before cutting it.

  Feedback: Cooling the chicken doesn't eliminate harmful bacteria. The cutting board could still become contaminated when used first to cut raw chicken and then vegetables without washing.

4.5. Sprint 5: Take a Quiz Functionality

In Sprint 5, we addressed the user stories presented in Table 15, focusing on the development of implementing functionality to enable chefs to take quizzes.

Table 15
Sprint 5 User Stories

Number	Title	User Type	I want to…	So that…
US1	View My Assigned Quizzes	Chef	View a list of quizzes assigned to me.	I can see which quizzes I need to complete.
US2	Access Past Quizzes	Chef	Review my answers.	I can reflect on my past performance.
US3	Take a Quiz	Chef	Take a quiz.	I can complete my assigned assessment tasks.
US4	Receive Feedback	Chef	Know my mark and feedback.	I can know if I passed and learn from my mistakes.

These user stories were then used to elaborate usability requirements, as shown in Table 16.

Table 16
Sprint 5 Usability Requirements

Number	Requirement	Description	User Story
UR1	Clear Presentation of User Quizzes	The chef should have access to a clear and organised list of quizzes assigned to them, showing relevant information such as completion status, deadline, and trainer.	US1
UR2	Easy Access to Past Quizzes	The chef should be able to access past quizzes and relevant information, such as the marks achieved, and the answers selected to each question.	US2
UR3	Intuitive Quiz Taking Interface	The experience of taking a quiz should be intuitive and straightforward.	US3
UR4	Access to Comprehensive Feedback	A chef should receive feedback upon completing a quiz, including a mark and an explanation of incorrect quiz question responses.	US4

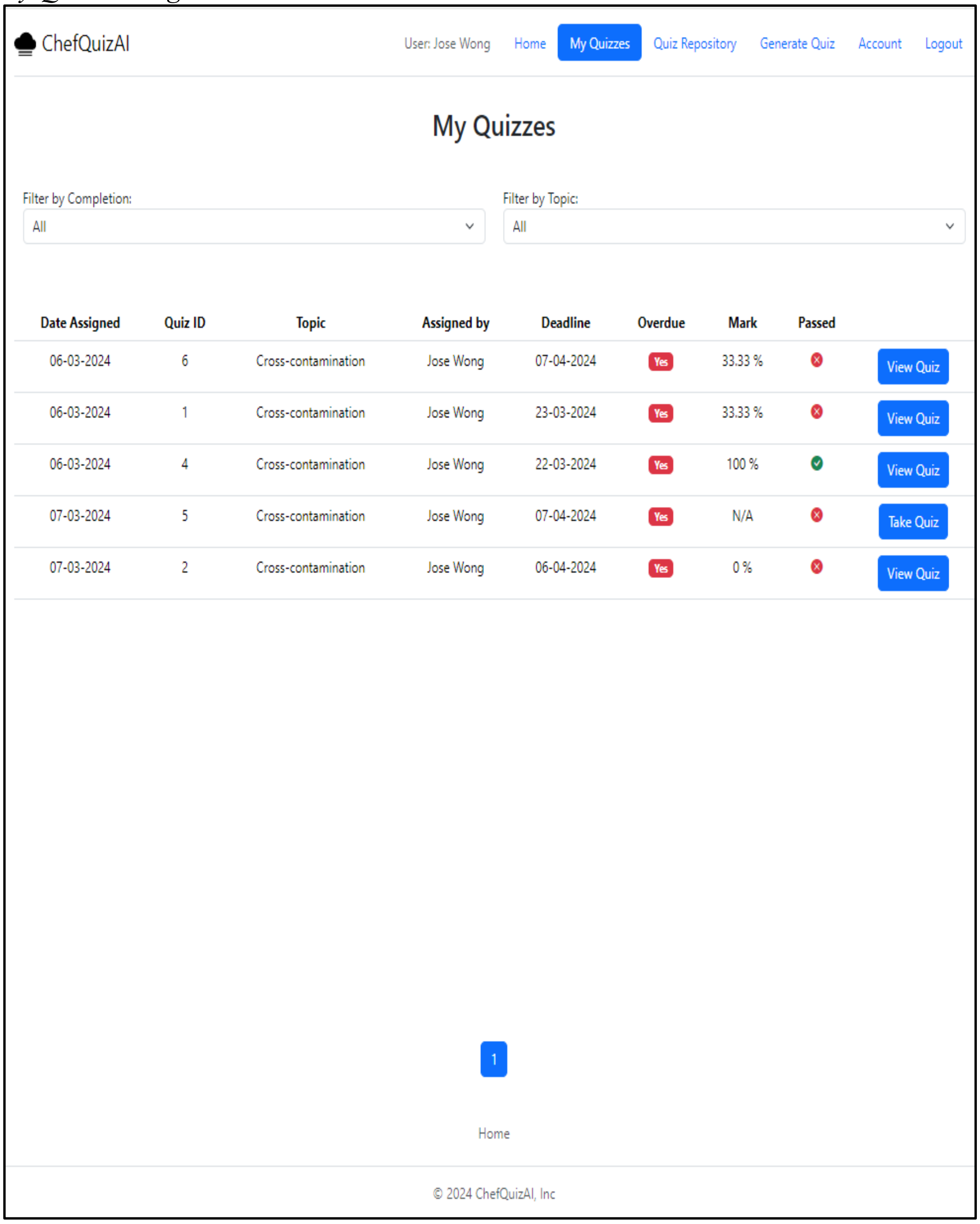
This list of usability requirements was then used to elaborate a list of the functionalities that the system was expected to provide, as shown in Table 17.

Table 17
Sprint 5 Functional Requirements

Number	Requirement	Description	User Requirements
FR1	Quiz Assignment Display	The system must provide the functionality to allow chefs to access a list of assigned quizzes they need to complete.	UR1
FR2	Past Quizzes Access	The system must implement functionality so that a chef can access past quizzes and review marks and responses.	UR2
FR3	Quiz Taking Interface	The system must implement functionality to allow chefs to take a quiz, including navigation buttons to move through the questions and prevent data loss during the process.	UR3
FR4	Feedback Delivery	The system must implement functionality to calculate a mark, retrieve appropriate feedback and present both upon quiz completion.	UR4

The approach to implement the requirements comprised the implementation of the "My Quizzes" page, designed using a Twig template, to display a list of quizzes assigned to a chef, including details such as the assignment date, quiz ID, topic, the trainer who assigned it, deadline, overdue status, achieved marks, pass status, and a button to interact with the quiz as shown in Figure 32. If that is the case, the button allows chefs to see the selected quiz for completion or the already completed quiz. For quizzes that require completion, the button opens a dedicated quiz-taking Twig template. In the template, implemented as a Bootstrap Carrousel, each card containing a question, as shown in Figure 33, counts with navigation buttons, allowing the chef to move forward to the next question, go back to the previous one, and view indicators of the current question number and remaining questions. Different sections of the card present the case scenario, the question itself and four selectable options via radio buttons.

Figure 32
My Quizzes Page



My Quizzes

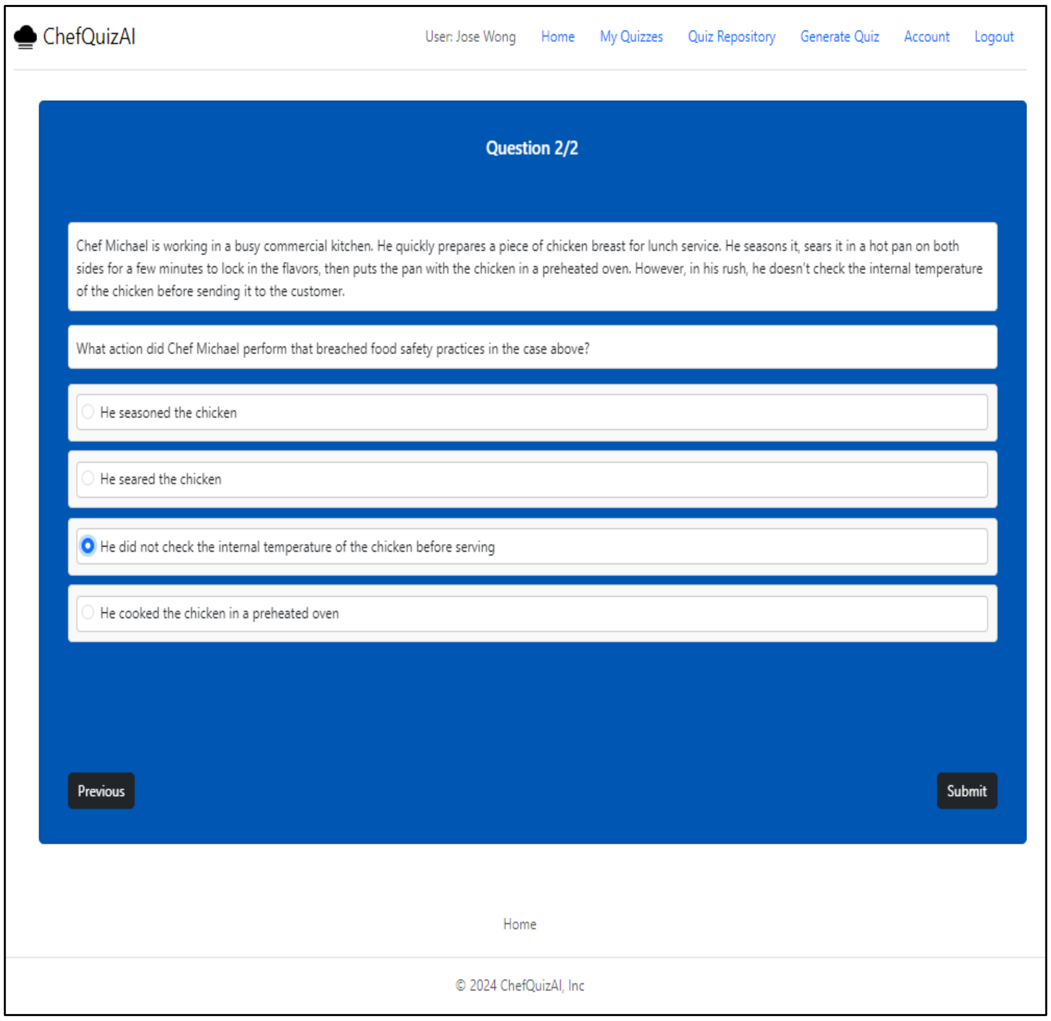
Filter by Completion: All

Filter by Topic: All

Date Assigned	Quiz ID	Topic	Assigned by	Deadline	Overdue	Mark	Passed	
06-03-2024	6	Cross-contamination	Jose Wong	07-04-2024	Yes	33.33 %	✗	View Quiz
06-03-2024	1	Cross-contamination	Jose Wong	23-03-2024	Yes	33.33 %	✗	View Quiz
06-03-2024	4	Cross-contamination	Jose Wong	22-03-2024	Yes	100 %	✓	View Quiz
07-03-2024	5	Cross-contamination	Jose Wong	07-04-2024	Yes	N/A	✗	Take Quiz
07-03-2024	2	Cross-contamination	Jose Wong	06-04-2024	Yes	0 %	✗	View Quiz

© 2024 ChefQuizAI, Inc

Figure 33
Quiz Taking Page



Question 2/2

Chef Michael is working in a busy commercial kitchen. He quickly prepares a piece of chicken breast for lunch service. He seasons it, sears it in a hot pan on both sides for a few minutes to lock in the flavors, then puts the pan with the chicken in a preheated oven. However, in his rush, he doesn't check the internal temperature of the chicken before sending it to the customer.

What action did Chef Michael perform that breached food safety practices in the case above?

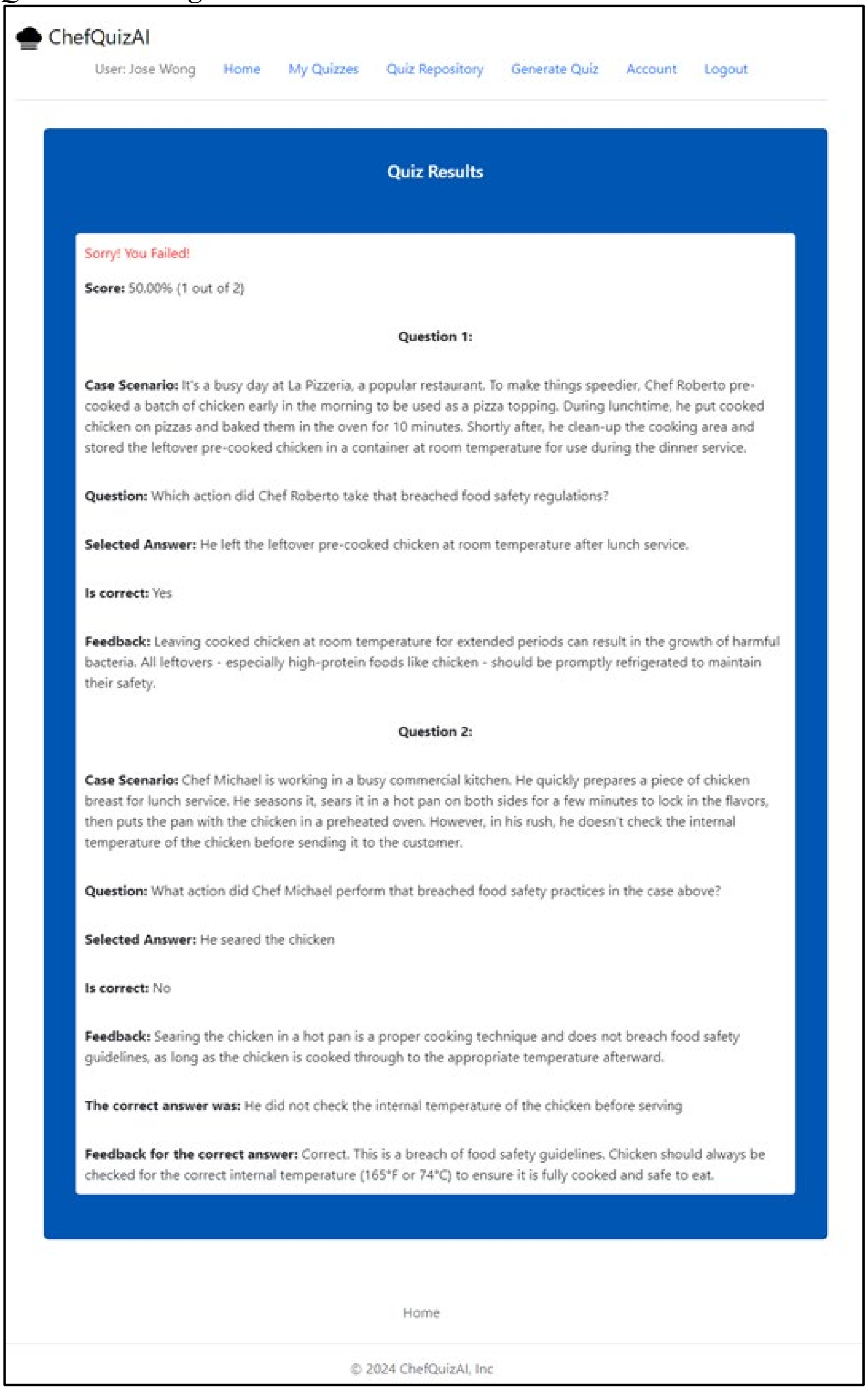
- ○ He seasoned the chicken
- ○ He seared the chicken
- ● He did not check the internal temperature of the chicken before serving
- ○ He cooked the chicken in a preheated oven

When taking a quiz, the corresponding controller fetches specific quiz details based on its ID and uses an implemented formatting service class to prepare the quiz for the user. If the quiz doesn't exist or the user is unauthorised, appropriate error messages are returned. Submitting a quiz involves the controller decoding the submitted answers from a JSON format, validating the user's right to submit the quiz, setting the quiz status to complete, calculating the user's score, saving the changes to the database, and returning a confirmation message.

The quiz-taking template is enhanced with a JavaScript script that controls the quiz-taking process on the front end. For instance, accidental reloads that could cause the inputted data to be lost and move to the next question without selecting an answer are not allowed.

At the end of the quiz, the script fetches the score and shows the result on the last card of the carrousel, as depicted in Figure 34. It gives detailed feedback for every question, according to the question selected by the user. At this point, it uses a POST request to finally submit the user's responses to the backend via the controller for recording.

Figure 34
Quiz Results Page



Quiz Results

Sorry! You Failed!

Score: 50.00% (1 out of 2)

Question 1:

Case Scenario: It's a busy day at La Pizzeria, a popular restaurant. To make things speedier, Chef Roberto pre-cooked a batch of chicken early in the morning to be used as a pizza topping. During lunchtime, he put cooked chicken on pizzas and baked them in the oven for 10 minutes. Shortly after, he clean-up the cooking area and stored the leftover pre-cooked chicken in a container at room temperature for use during the dinner service.

Question: Which action did Chef Roberto take that breached food safety regulations?

Selected Answer: He left the leftover pre-cooked chicken at room temperature after lunch service.

Is correct: Yes

Feedback: Leaving cooked chicken at room temperature for extended periods can result in the growth of harmful bacteria. All leftovers - especially high-protein foods like chicken - should be promptly refrigerated to maintain their safety.

Question 2:

Case Scenario: Chef Michael is working in a busy commercial kitchen. He quickly prepares a piece of chicken breast for lunch service. He seasons it, sears it in a hot pan on both sides for a few minutes to lock in the flavors, then puts the pan with the chicken in a preheated oven. However, in his rush, he doesn't check the internal temperature of the chicken before sending it to the customer.

Question: What action did Chef Michael perform that breached food safety practices in the case above?

Selected Answer: He seared the chicken

Is correct: No

Feedback: Searing the chicken in a hot pan is a proper cooking technique and does not breach food safety guidelines, as long as the chicken is cooked through to the appropriate temperature afterward.

The correct answer was: He did not check the internal temperature of the chicken before serving

Feedback for the correct answer: Correct. This is a breach of food safety guidelines. Chicken should always be checked for the correct internal temperature (165°F or 74°C) to ensure it is fully cooked and safe to eat.

Home

© 2024 ChefQuizAI, Inc

4.6. Sprint 6: Prompt Refinement and Retrieval Augmented Generation

This sprint was focused on improving the relevance, level of detail and accuracy of the responses given from the integrated GPT-4 API service. This improvement was very important in assuring relevance to the food safety training requirements guided by the expectations demonstrated in the relevant user stories presented in Table 18.

Table 18
Sprint 6 User Stories

Number	Title	User Type	I want to…	So that…
US1	Precise Quiz Generation	Trainer	Generate quizzes with accurate and precise food safety content.	I can test chefs' food safety knowledge and assess further training needs.
US2	Enhanced Quiz-Taking	Chef	Be presented with engaging and challenging food safety quizzes.	I can assess and improve my knowledge of food safety knowledge and awareness.

Consequently, these user stories were turned into corresponding usability requirements, as shown in Table 19

Table 19
Sprint 6 Usability Requirements

Number	Requirement	Description	User Story
UR1	Easily Generate Precise Quiz Content	A Trainer should be able to compile quizzes with relevant, precise, accurate, and detailed question content.	US1
UR2	Fulfilling Quiz-Taking Experience	A chef should be able to take a quiz with relevant, precise, accurate and detailed content.	US2

This list of usability requirements was then represented as a functional requirement that the system must provide, as presented in Table 20.

Table 20
Sprint 6 Functional Requirements

Number	Requirement	Description	User Requirement
FR1	Enhanced Question Generation Service	The system must implement functionality to generate relevant, accurate, precise and detailed food safety best practices information quiz questions.	UR1, UR2

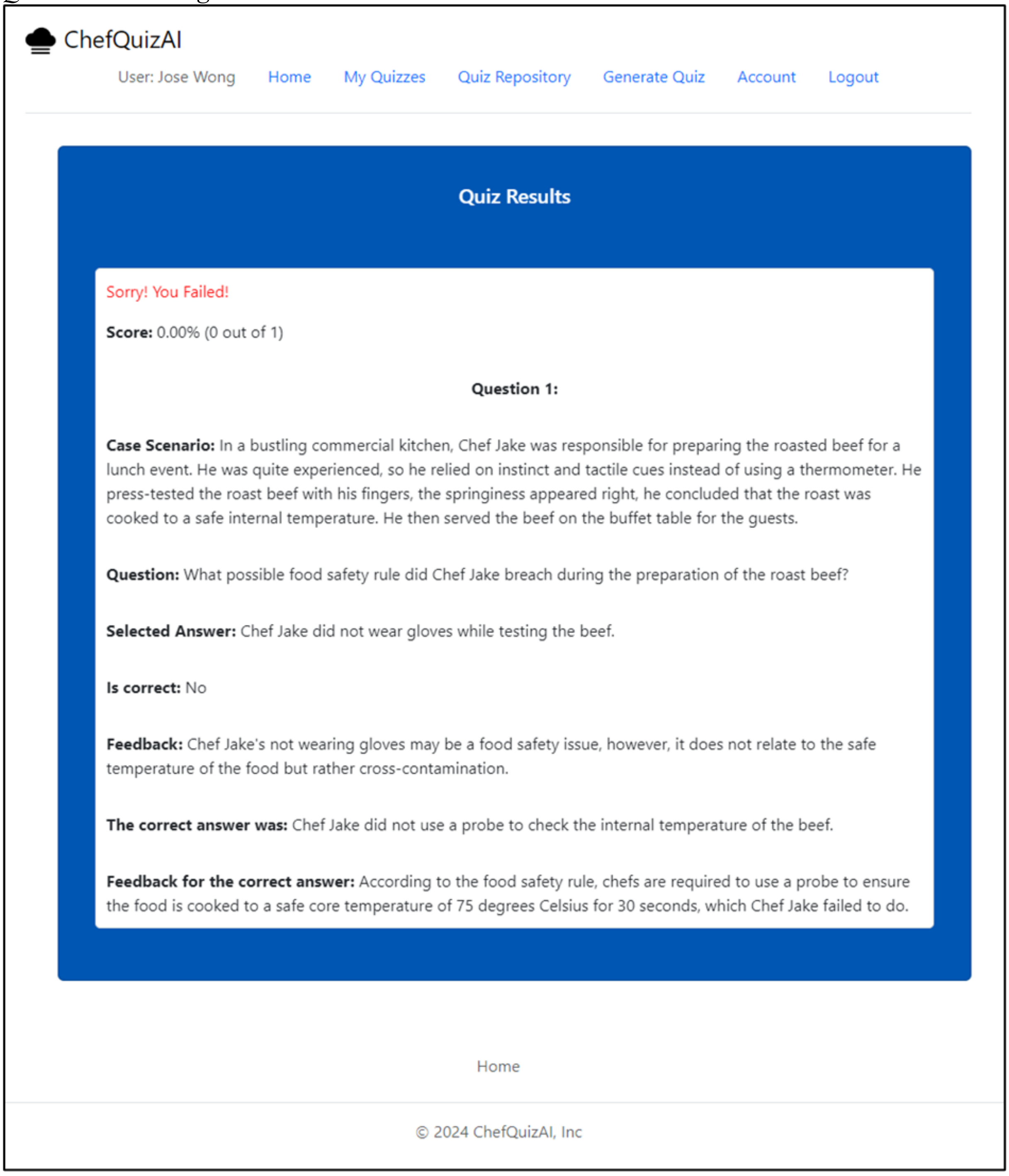
The first approach to solving the requirement was revising the prompt to provide an example of the type of response expected from the GPT-4 API. The purpose was to guide the AI in producing outputs that come close to the quality of the structure and content expected so that irrelevant or wrong information is minimised. This approach aligns with "zero-shot" prompt engineering, a form of example-based learning where an example of expected output is provided to guide the AI's response generation. (Mizrahi, 2023).

Additionally, a new entity, FoodSafetyBestPractice, was added to the data model. The corresponding database table was created to store a custom repository containing entries related to food safety good practices. The entries in this table were aligned with various topics within the realm of food safety. When forming the GPT-4 API prompt, the system randomly selects relevant entries from this database according to the topic parameter of the quiz question. The selected food safety good practice is injected directly into the prompt, and the entries become reference points for the AI to ensure relevance and accuracy are on point. As a result of these improvements, the generated content saw a significant increase in detail. The replies from the GPT-4 API began to include more specific information, such as temperatures required by law for cooking different kinds of foods, which are needed in safety training.

For instance, after including the following entry in the FoodSafetyBestPractice table: "*You must use a food probe to ensure any food has been cooked until it reaches a combination of safe core temperature and required length of time. For instance, a safe core temperature is 75 degrees Celsius for 30 seconds*." as demonstrated in Figure 35, the system could produce the following accurate feedback in the generated question: "*According to the food safety rule, chefs are required to use a probe to ensure the food is cooked to a safe core temperature of 75 degrees Celsius for 30 seconds, which Chef Jake failed to do*.", which included a very specific legal requirement of internal temperature food must reach when cooking. Consequently, the quiz taker received precise feedback when selecting the wrong answer for the question.

The FoodSafetyBestPractice repository was also used to act on a request for functionality to produce entry-level quizzes. The approach was to create a new topic, "Initial Food Safety Training", in the frontend functionality to generate quizzes and an additional set of entries in the FoodSafetyBestPractice table using "Entry Level" as the topic. These entries in the table consisted of basic food safety practices that a new employee should be briefed on upon starting employment. Consequently, when a new quiz is created, and this topic is selected, the backend quiz generation service injects it into the prompt and requests the language model for the generation of questions accordingly, allowing for the creation of entry-level quizzes. The same approach could be used to generate different levels of quizzes if the application were adapted for other use cases.

Figure 35
Quiz Feedback Page



Quiz Results

Sorry! You Failed!

Score: 0.00% (0 out of 1)

Question 1:

Case Scenario: In a bustling commercial kitchen, Chef Jake was responsible for preparing the roasted beef for a lunch event. He was quite experienced, so he relied on instinct and tactile cues instead of using a thermometer. He press-tested the roast beef with his fingers, the springiness appeared right, he concluded that the roast was cooked to a safe internal temperature. He then served the beef on the buffet table for the guests.

Question: What possible food safety rule did Chef Jake breach during the preparation of the roast beef?

Selected Answer: Chef Jake did not wear gloves while testing the beef.

Is correct: No

Feedback: Chef Jake's not wearing gloves may be a food safety issue, however, it does not relate to the safe temperature of the food but rather cross-contamination.

The correct answer was: Chef Jake did not use a probe to check the internal temperature of the beef.

Feedback for the correct answer: According to the food safety rule, chefs are required to use a probe to ensure the food is cooked to a safe core temperature of 75 degrees Celsius for 30 seconds, which Chef Jake failed to do.

© 2024 ChefQuizAI, Inc

4.7. Sprint 7 User Interface Refinement

This sprint addressed users' feedback from the previous sprint regarding the user interface navigation and browsing requirements, as represented by the user stories in Table 21.

Table 21
Sprint 7 User Stories

Number	Title	User Type	I want to…	So that…
US1	Quiz Search	All Users	Filter large lists of quizzes.	I can easily find the quizzes that I need.
US2	Quick Access to Quizzes Lists	All Users	Access large lists of quizzes quickly.	I don't have to wait for long lists to load.

Subsequently, these user stories were converted into a set of usability requirements presented in Table 22.

Table 22
Sprint 7 Usability Requirements

Number	Requirement	Description	User Story
UR1	Filter Quizzes Lists	A user should be able to easily filter large lists of quizzes based on multiple criteria such as topic and completion status.	US1
UR2	Quizzes' Lists Sequential Loading	A user should be able to quickly be presented with information in a large list of quizzes without waiting for the entire list to load.	US2

The usability requirements were then used to elaborate the functional requirements that the application needed to implement.

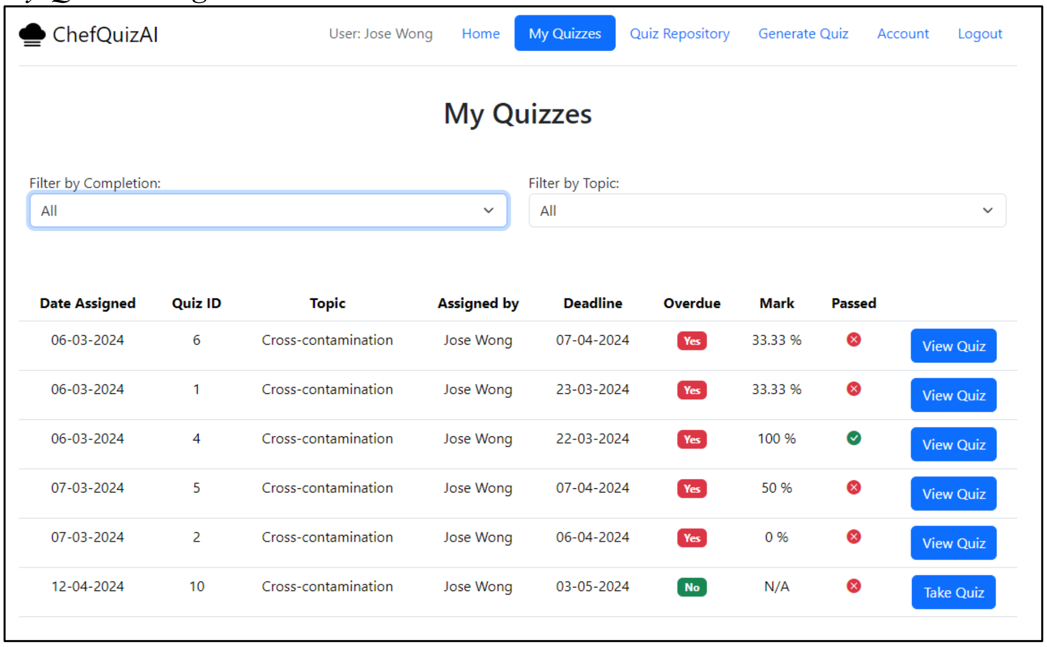
Table 23
Sprint 7 Functional Requirements

Number	Requirement	Description	User Requirement
FR1	Dropdown Selection Filters	The application must provide dropdown select elements and corresponding queries to retrieve quizzes based on the inputted select criteria.	UR1
FR1	Pagination	The application must provide pagination buttons and navigation arrows to navigate across lists of quizzes and fetch information sequentially as the pages are navigated.	UR1

The solution to the filtering requirements was to implement dropdown menus to select a trainer name and a topic in the Quiz Repository page and similar elements to select the completion status and topic in the My Quizzes page. JavaScript functions were used to request data based on the selected criteria via the corresponding controllers, which fetch the data using custom methods in the corresponding Quiz and AssignedQuiz entities repositories. Figure 36 shows the filters implemented on the My Quizzes page.

Figure 36

My Quizzes Page Filters



My Quizzes

Filter by Completion: All

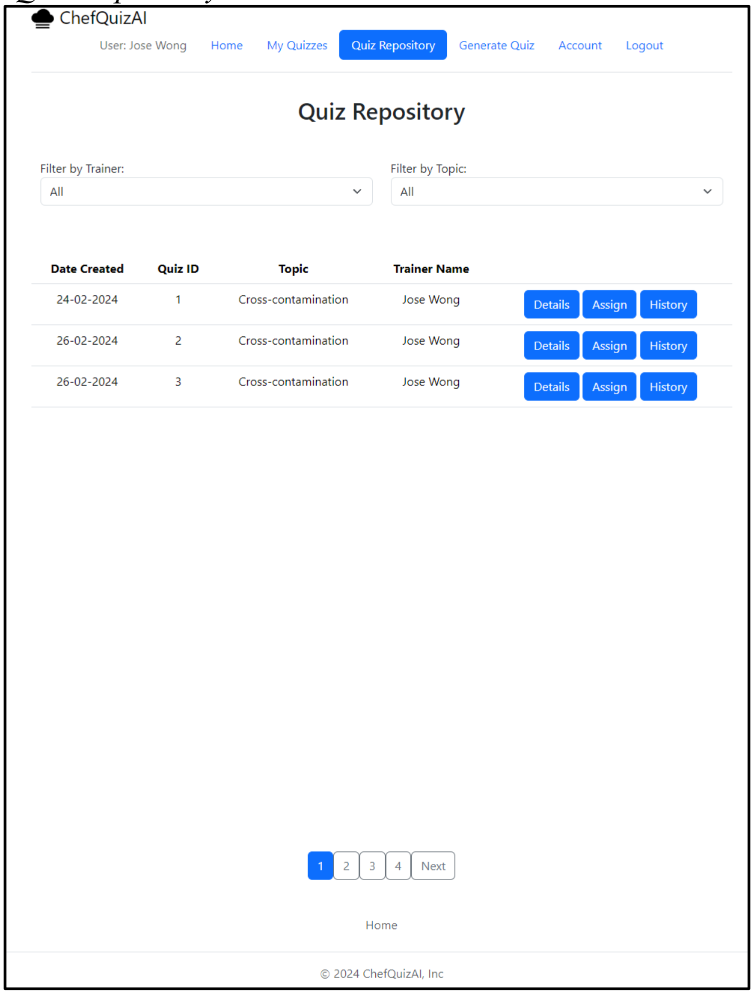
Filter by Topic: All

Date Assigned	Quiz ID	Topic	Assigned by	Deadline	Overdue	Mark	Passed	
06-03-2024	6	Cross-contamination	Jose Wong	07-04-2024	Yes	33.33 %	✗	View Quiz
06-03-2024	1	Cross-contamination	Jose Wong	23-03-2024	Yes	33.33 %	✗	View Quiz
06-03-2024	4	Cross-contamination	Jose Wong	22-03-2024	Yes	100 %	✓	View Quiz
07-03-2024	5	Cross-contamination	Jose Wong	07-04-2024	Yes	50 %	✗	View Quiz
07-03-2024	2	Cross-contamination	Jose Wong	06-04-2024	Yes	0 %	✗	View Quiz
12-04-2024	10	Cross-contamination	Jose Wong	03-05-2024	No	N/A	✗	Take Quiz

The implemented solution to solve the pagination requirement uses Symfony's PaginatorInterface in the controller to fetch a slice of the required data at a time, which is then passed to the Twig template, where a JavaScript function dynamically updates the pagination controls, including highlighting the numeric button corresponding to the current page and adding next or previous buttons as needed. Figure 37 shows the pagination controls for the Quiz Repository page.

Figure 37

Quiz Repository Filters



Quiz Repository

Filter by Trainer: All

Filter by Topic: All

Date Created	Quiz ID	Topic	Trainer Name	
24-02-2024	1	Cross-contamination	Jose Wong	Details Assign History
26-02-2024	2	Cross-contamination	Jose Wong	Details Assign History
26-02-2024	3	Cross-contamination	Jose Wong	Details Assign History

© 2024 ChefQuizAI, Inc

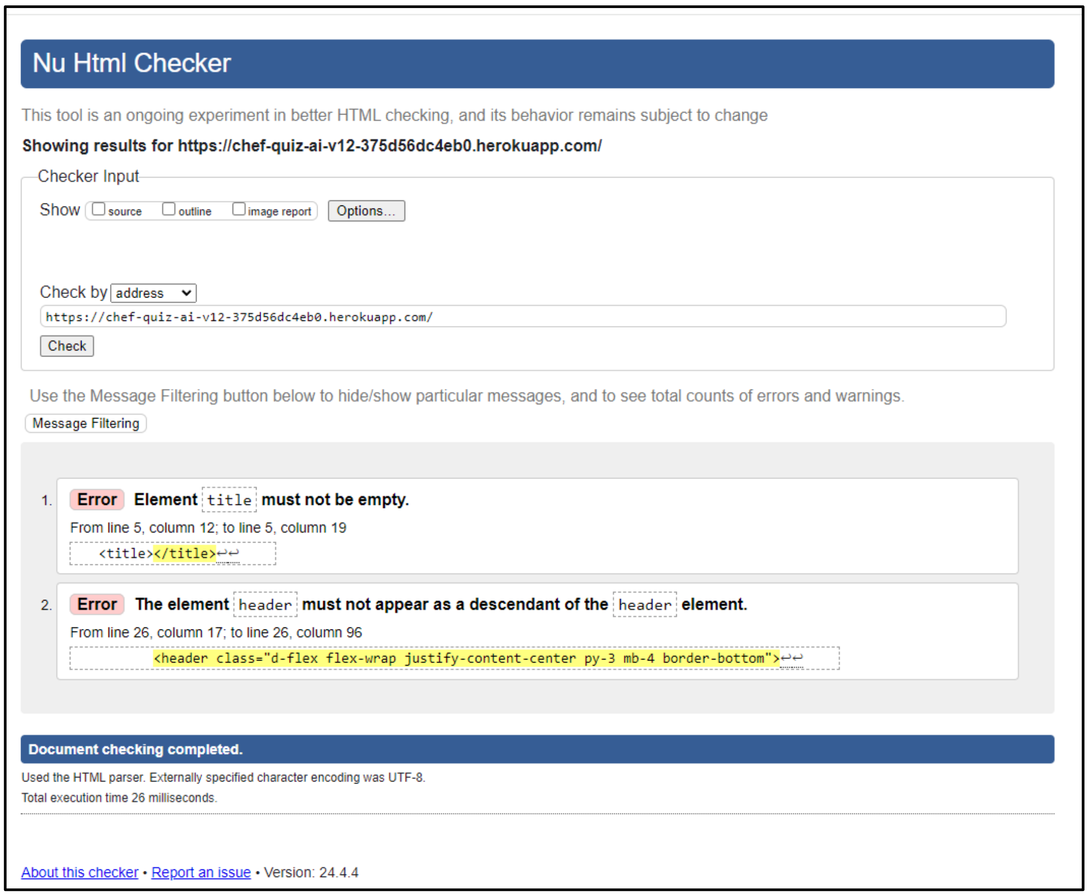
Chapter 5. Testing and Results

The test plan included the use of PHPUnit unit tests. Test files were added during the corresponding sprints to validate these implementations. Additionally, functional testing, accessibility testing, HTML & CSS validation, cross-browser, and device independence testing were carried out along the development process during the sprints in accordance with the methods and approaches described in the sections of this chapter.

5.1. HTML & CSS Validation

We used the W3C validator to test HTML and CSS correctness. Figure 38 shows how this could help identify errors, such as a duplicated header element in the application's layout template, which was subsequently corrected.

Figure 38
W3C HTML and CSS Validation



Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Checker Input

Show ☐ source ☐ outline ☐ image report Options...

Check by address

`https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/`

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Error Element `title` must not be empty.**
   From line 5, column 12; to line 5, column 19
   ```
   <title></title>↩↩
   ```
2. **Error The element `header` must not appear as a descendant of the `header` element.**
   From line 26, column 17; to line 26, column 96
   ```
   <header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">↩↩
   ```

Document checking completed.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 26 milliseconds.

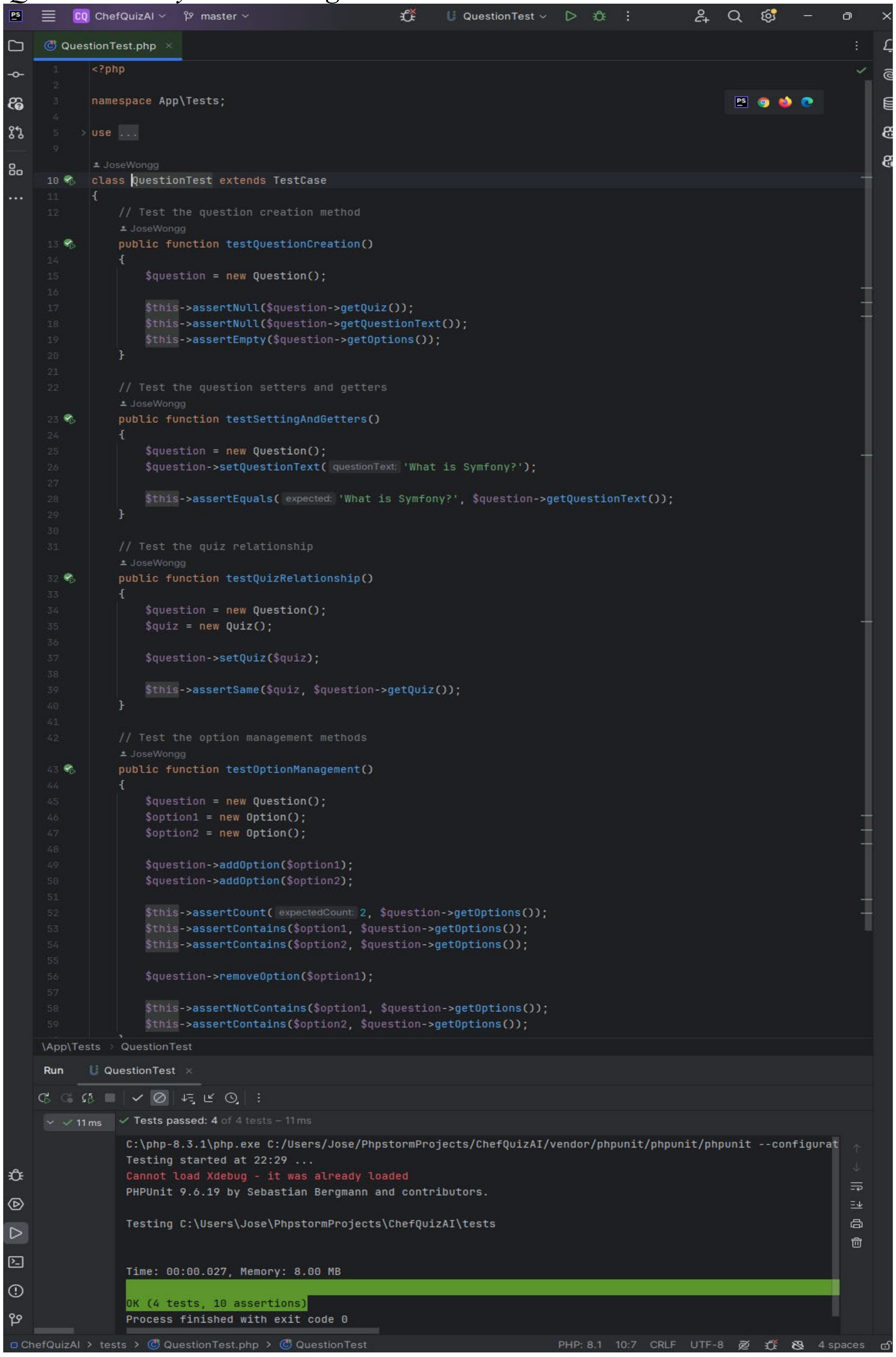
About this checker • Report an issue • Version: 24.4.4

5.2. Unit Testing

We conduct unit testing to verify that the individual components or units of code in an application function correctly in isolation. This testing was executed for the User, Quiz, Question, Option, AssignedQuiz, and FoodSafetyBestPractices Entities methods. Figure 39 shows an example of these tests.

Figure 39

Question Entity Unit Testing



```php
<?php

namespace App\Tests;

use ...

JoseWongg
class QuestionTest extends TestCase
{
    // Test the question creation method
    JoseWongg
    public function testQuestionCreation()
    {
        $question = new Question();

        $this->assertNull($question->getQuiz());
        $this->assertNull($question->getQuestionText());
        $this->assertEmpty($question->getOptions());
    }

    // Test the question setters and getters
    JoseWongg
    public function testSettingAndGetters()
    {
        $question = new Question();
        $question->setQuestionText( questionText: 'What is Symfony?');

        $this->assertEquals( expected: 'What is Symfony?', $question->getQuestionText());
    }

    // Test the quiz relationship
    JoseWongg
    public function testQuizRelationship()
    {
        $question = new Question();
        $quiz = new Quiz();

        $question->setQuiz($quiz);

        $this->assertSame($quiz, $question->getQuiz());
    }

    // Test the option management methods
    JoseWongg
    public function testOptionManagement()
    {
        $question = new Question();
        $option1 = new Option();
        $option2 = new Option();

        $question->addOption($option1);
        $question->addOption($option2);

        $this->assertCount( expectedCount: 2, $question->getOptions());
        $this->assertContains($option1, $question->getOptions());
        $this->assertContains($option2, $question->getOptions());

        $question->removeOption($option1);

        $this->assertNotContains($option1, $question->getOptions());
        $this->assertContains($option2, $question->getOptions());
```

\App\Tests › QuestionTest

✓ Tests passed: 4 of 4 tests – 11 ms

```
C:\php-8.3.1\php.exe C:/Users/Jose/PhpstormProjects/ChefQuizAI/vendor/phpunit/phpunit/phpunit --configurat
Testing started at 22:29 ...
Cannot load Xdebug - it was already loaded
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

Testing C:\Users\Jose\PhpstormProjects\ChefQuizAI\tests


Time: 00:00.027, Memory: 8.00 MB

OK (4 tests, 10 assertions)
Process finished with exit code 0
```

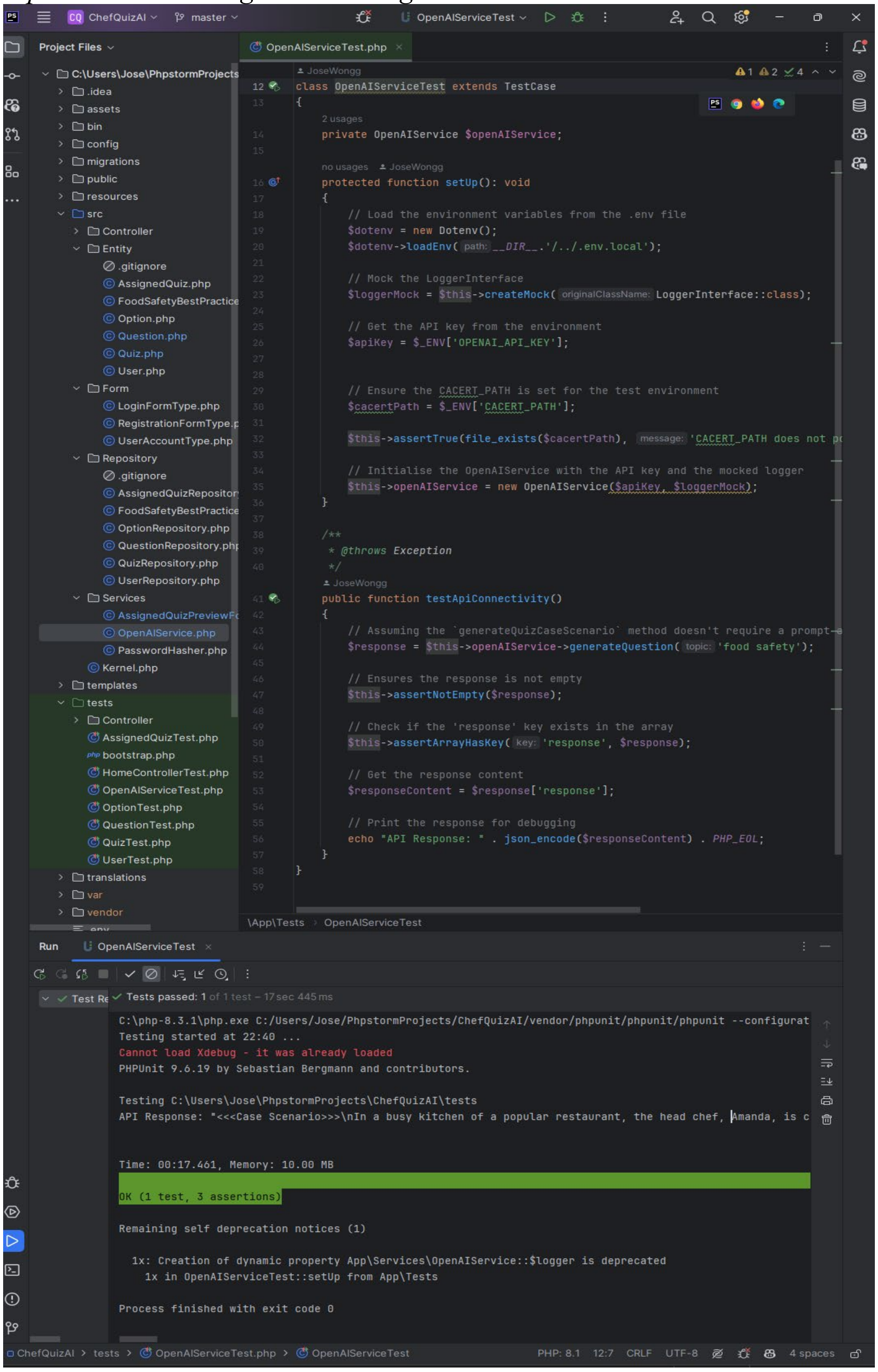
ChefQuizAI › tests › QuestionTest.php › QuestionTest   PHP: 8.1 10:7 CRLF UTF-8 4 spaces

5.3. Integration Testing

A PHPUnit integration test was implemented to test the integration of the application with the OpenAI API, as shown in Figure 40.

Figure 40

OpenAI Service Integration Testing



```php
class OpenAIServiceTest extends TestCase
{
    private OpenAIService $openAIService;

    protected function setUp(): void
    {
        // Load the environment variables from the .env file
        $dotenv = new Dotenv();
        $dotenv->loadEnv( path: __DIR__.'/../.env.local');

        // Mock the LoggerInterface
        $loggerMock = $this->createMock( originalClassName: LoggerInterface::class);

        // Get the API key from the environment
        $apiKey = $_ENV['OPENAI_API_KEY'];


        // Ensure the CACERT_PATH is set for the test environment
        $cacertPath = $_ENV['CACERT_PATH'];

        $this->assertTrue(file_exists($cacertPath), message: 'CACERT_PATH does not po

        // Initialise the OpenAIService with the API key and the mocked logger
        $this->openAIService = new OpenAIService($apiKey, $loggerMock);
    }

    /**
     * @throws Exception
     */
    public function testApiConnectivity()
    {
        // Assuming the `generateQuizCaseScenario` method doesn't require a prompt-a
        $response = $this->openAIService->generateQuestion( topic: 'food safety');

        // Ensures the response is not empty
        $this->assertNotEmpty($response);

        // Check if the 'response' key exists in the array
        $this->assertArrayHasKey( key: 'response', $response);

        // Get the response content
        $responseContent = $response['response'];

        // Print the response for debugging
        echo "API Response: " . json_encode($responseContent) . PHP_EOL;
    }
}
```

Tests passed: 1 of 1 test – 17 sec 445 ms

```
C:\php-8.3.1\php.exe C:/Users/Jose/PhpstormProjects/ChefQuizAI/vendor/phpunit/phpunit/phpunit --configurat
Testing started at 22:40 ...
Cannot load Xdebug - it was already loaded
PHPUnit 9.6.19 by Sebastian Bergmann and contributors.

Testing C:\Users\Jose\PhpstormProjects\ChefQuizAI\tests
API Response: "<<<Case Scenario>>>\nIn a busy kitchen of a popular restaurant, the head chef, Amanda, is c


Time: 00:17.461, Memory: 10.00 MB

OK (1 test, 3 assertions)

Remaining self deprecation notices (1)

  1x: Creation of dynamic property App\Services\OpenAIService::$logger is deprecated
    1x in OpenAIServiceTest::setUp from App\Tests

Process finished with exit code 0
```

5.4. Functional Testing

Functional testing was performed during each sprint to validate the implemented functionalities against the functional requirements, as presented in Tables 24 to 30.

Table 24
Sprint 1 Functional Testing

Sprint 1 Functional Requirements	Status
The system must allow users to access an entry point through a homepage without downloading or installing software.	Passed
The system must allow the user to navigate from one page to another.	Passed
The system must allow users to view content responsively in different screen sizes.	Passed
The system must allow the user to access accurate data and guarantee its integrity.	Passed
The system must ensure the user can access the latest version without intervention.	Passed

Table 25
Sprint 2 Functional Testing

Sprint 2 Functional Requirements	Status
The system must verify login credentials against the database and provide access according to the user-predefined roles.	Passed
The system must display appropriate errors when users enter wrong input during registration, account update and logging.	Passed
The system must allow users to terminate sessions, clear session data, and redirect users to the homepage.	Passed
The system must provide functionality to register new users and store their profiles in the database.	Passed
The system must hash new passwords and store them securely.	Passed
The system must allow users to update their profile details and safely store changes.	Passed
The system must provide confirmation when the user logs in, registers, and changes their profile details.	Passed

Table 26
Sprint 3 Functional Testing

Sprint 3 Functional Requirements	Status
The system must provide a service to generate questions automatically based on selected topics.	Passed
The system must implement functionality to edit generated questions.	Passed
The system must provide an area to visualise a question and a button to approve it.	Passed
The system must provide a button to delete a question.	Passed
The system must provide the functionality to preview a quiz.	Passed
The system must provide a dropdown list to select a topic and a button to generate a question.	Passed

Table 27
Sprint 4 Functional Testing

Sprint 4 Functional Requirements	Status
The system must provide a user interface for interaction and information about a quiz that can be assigned to a chef.	Passed
The system must implement functionality to assign a quiz to a chef or group of chefs, setting a passing mark and deadline.	Passed
The system must implement an interface to provide information about past quizzes already taken by any user.	Passed

Table 28
Sprint 5 Functional Testing

Sprint 5 Functional Requirements	Status
The system must provide the functionality to allow chefs to access a list of assigned quizzes that they need to complete.	Passed
The system must implement functionality so that a chef can access past quizzes and review marks and responses.	Passed
The system must implement functionality to allow chefs to take a quiz, including navigation buttons to move through the questions and prevent data loss during the process.	Passed
The system must implement functionality to calculate a mark, retrieve appropriate feedback and present both upon quiz completion.	Passed

Table 29
Sprint 6 Functional Testing

Sprint 6 Functional Requirement	Status
The system must implement functionality to generate relevant, accurate, precise, and detailed food safety best practices information quiz questions.	Passed

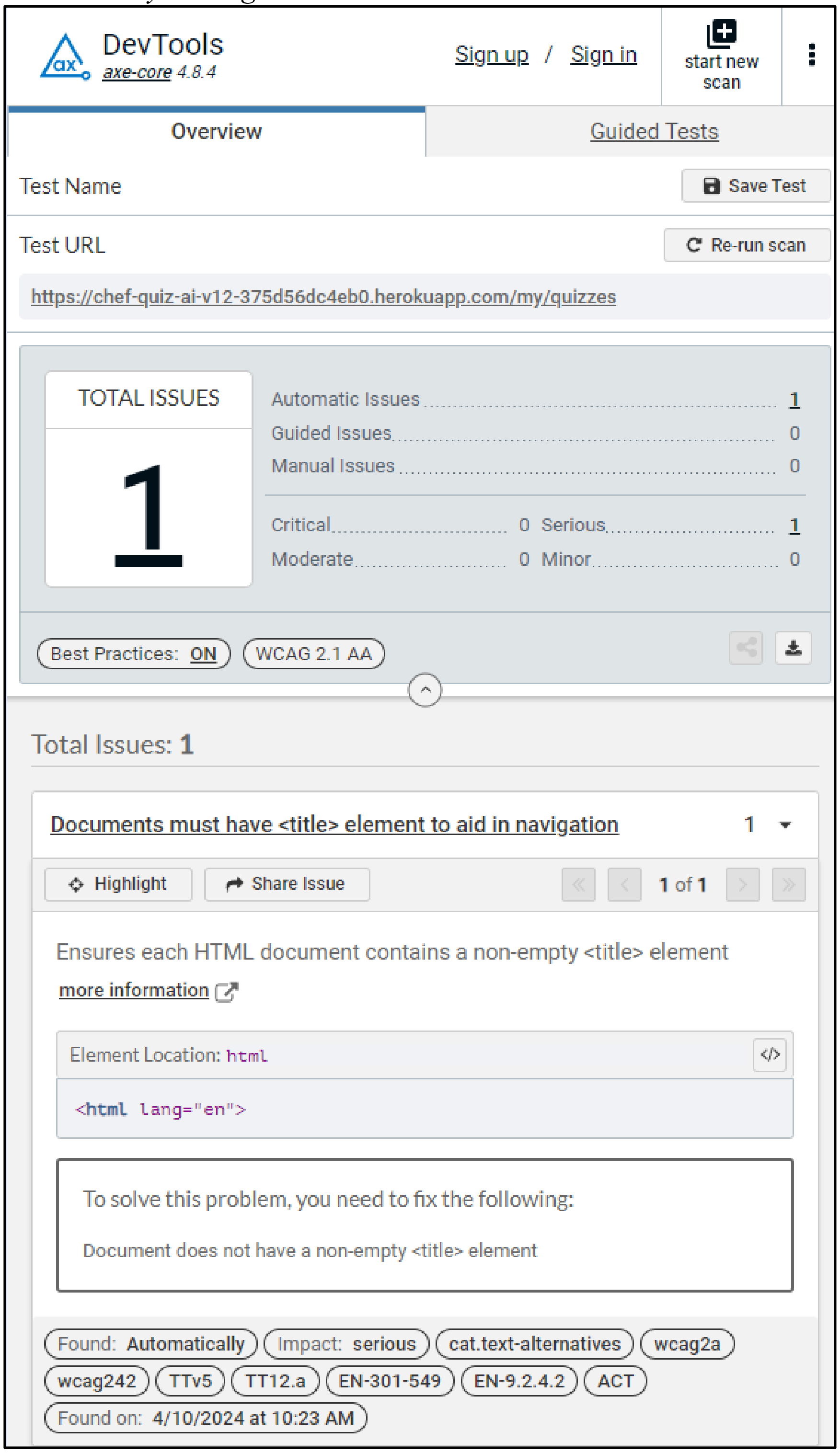
Table 30
Sprint 7 Functional Testing

Sprint 7 Functional Requirement	Status
The application must provide dropdown select elements and corresponding queries to retrieve quizzes based on the inputted select criteria.	Passed
The application must provide pagination buttons and navigation arrows to navigate across lists of quizzes and fetch information sequentially as the pages are navigated.	Passed

5.5. Accessibility Testing

For accessibility testing, we used the tool axe DevTools. This browser extension scanned the application's web pages to highlight issues due to a lack of compliance with the Web Content Accessibility Guidelines (WCAG) 2.1AA standards, the benchmark for developers to create web content that individuals with visual, auditory, motor, or cognitive impairments can access. This helped identify an issue, as shown in Figure 41, related to a missing <title> element in the application's base template, used by assistive technology such as screen readers, which was subsequently fixed.

Figure 41
Accessibility Testing


DevTools axe-core 4.8.4
Sign up / Sign in
start new scan
Overview | Guided Tests
Test Name — Save Test
Test URL — Re-run scan
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/my/quizzes

TOTAL ISSUES 1

Automatic Issues	1
Guided Issues	0
Manual Issues	0
Critical	0
Serious	1
Moderate	0
Minor	0

Best Practices: ON | WCAG 2.1 AA

Total Issues: 1

Documents must have <title> element to aid in navigation 1

Highlight | Share Issue | 1 of 1

Ensures each HTML document contains a non-empty <title> element
more information

Element Location: html

```
<html lang="en">
```

To solve this problem, you need to fix the following:

Document does not have a non-empty <title> element

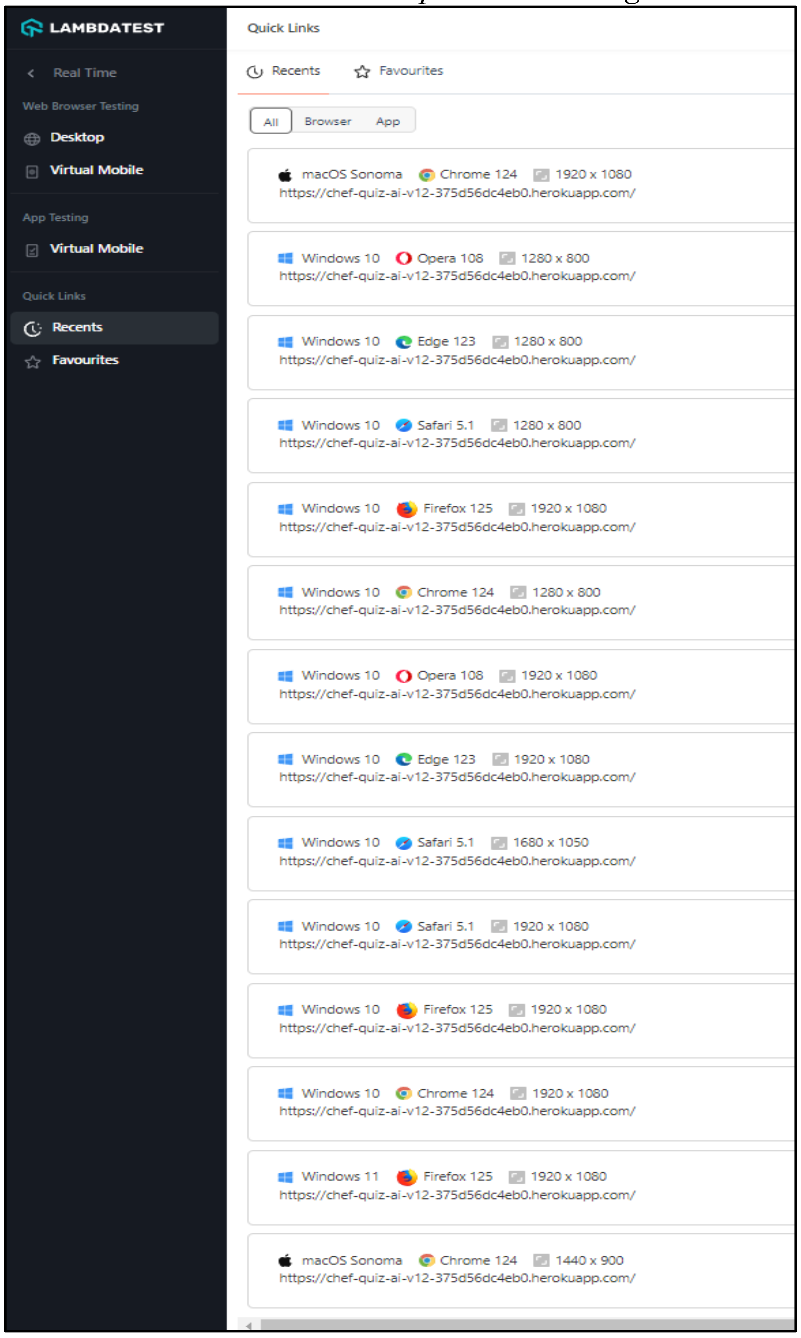
Found: Automatically | Impact: serious | cat.text-alternatives | wcag2a | wcag242 | TTv5 | TT12.a | EN-301-549 | EN-9.2.4.2 | ACT
Found on: 4/10/2024 at 10:23 AM

5.6. Cross Browser and Device Independence Testing

For cross-browser and device independence tests, the Lambdatest online platform, as shown in Figure 42, allowed us to test the application using several browsers, including Chrome, Firefox, Safari, Edge, and Opera, and simulate Windows 10, Windows 11, and MacOS Sonoma environments at full HD 1920x1080 and 1024x768 for desktop and 10-inch tablet devices, respectively. No single issues were found in this testing in terms of responsiveness, navigation, and functionality. Following this approach, we could verify that the application behaves as expected in all these environments.

Figure 42

Cross Browser and Device Independence Testing



LAMBDATEST

Real Time

Web Browser Testing

Desktop

Virtual Mobile

App Testing

Virtual Mobile

Quick Links

Recents

Favourites

Quick Links

Recents Favourites

All Browser App

macOS Sonoma Chrome 124 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Opera 108 1280 x 800
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Edge 123 1280 x 800
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Safari 5.1 1280 x 800
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Firefox 125 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Chrome 124 1280 x 800
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Opera 108 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Edge 123 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Safari 5.1 1680 x 1050
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Safari 5.1 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Firefox 125 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 10 Chrome 124 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

Windows 11 Firefox 125 1920 x 1080
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

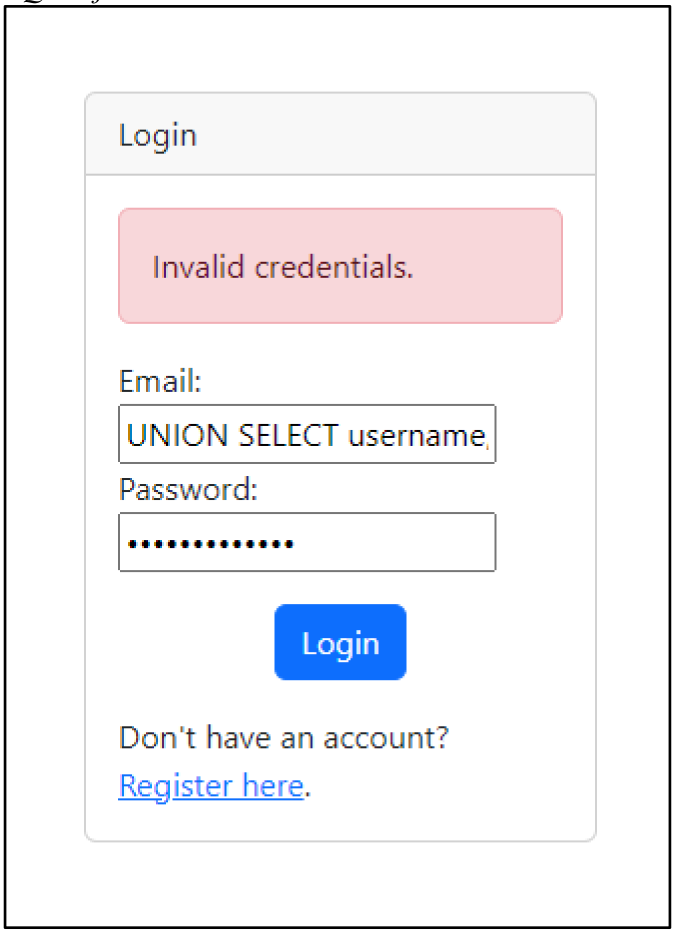
macOS Sonoma Chrome 124 1440 x 900
https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/

5.7. Penetration Testing

The use of the MVC pattern within the Symfony framework plays a significant role in protecting the application against malicious attacks and avoiding the use of SQL database queries in the controller. Hence, data is not manipulated at this level but handled by the model layer entities and repositories using Doctrine ORM for database interactions. This implies an abstraction layer that separates SQL queries from data using prepared statements and parameterised queries with placeholders instead of user input, providing solid protection against SQL injection attacks.

To test the efficiency of this approach, we prepared a query: ' UNION SELECT username, password FROM users –' and exerted an attack by injecting it into every publicly accessible input field, including the registration form and the login form. This attack is intended to cause the application to retrieve login credentials from the user's table and dump them in its response, which could be used for unauthorised log in. As demonstrated in Figure 43, the application rejected the attempt, and the developer's console tools did not show any response.

Figure 43
SQL Injection Attack



Login

Invalid credentials.

Email:

UNION SELECT username,

Password:

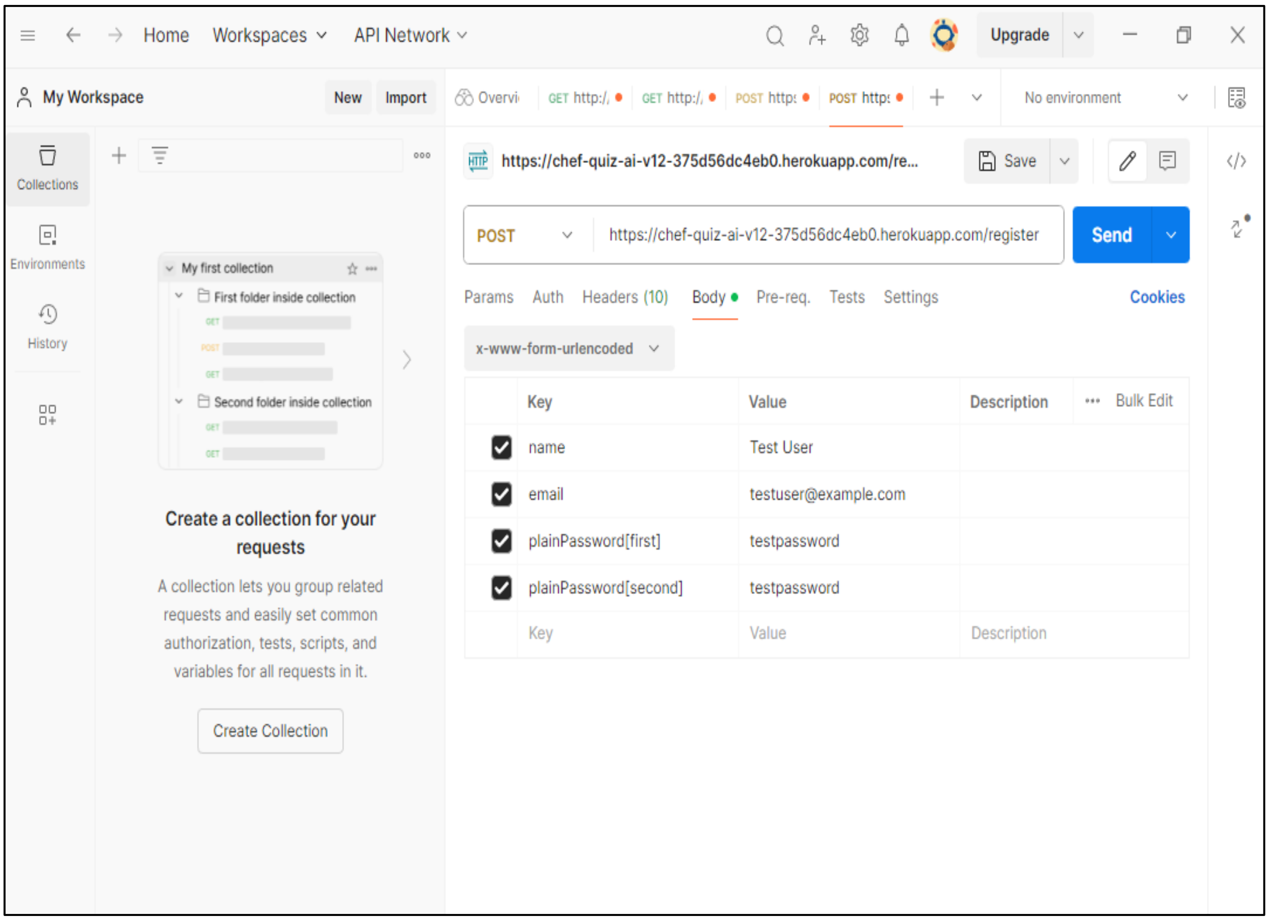
•••••••••••••

Don't have an account?
Register here.

Another common technique to compromise a website, known as Cross-Site Request Forgery (CSRF), involves an attacker's attempt to exploit a user's authenticated session to perform actions on their behalf. For instance, targeting online banking application users with the purpose of changing their login credentials or even carrying out money transfers while the users are logged in. Symfony has CSRF protection enabled by default, and its forms contain a hidden field. Symfony's security mechanism fills this field with a CSRF Token associated with the user session so that only the user can submit the form. To test this mechanism, we simulated a CSRF attack by attempting to submit a change of login credentials form in the application using Postman. This software testing tool allows sending requests, as shown in Figure 44. After submitting the request, we analysed the response in the Symfony profiler tool, where it was clear that the request was rejected due to a missing CSRF token, as demonstrated in Figure 45.

Figure 44

Cross-Site Request Forgery Test


Home
Workspaces
API Network
Upgrade
My Workspace
New
Import
No environment
Collections
Environments
History
My first collection
First folder inside collection
Second folder inside collection
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
Save
Send
Params
Auth
Headers (10)
Body
Pre-req.
Tests
Settings
Cookies
Bulk Edit

https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/re...

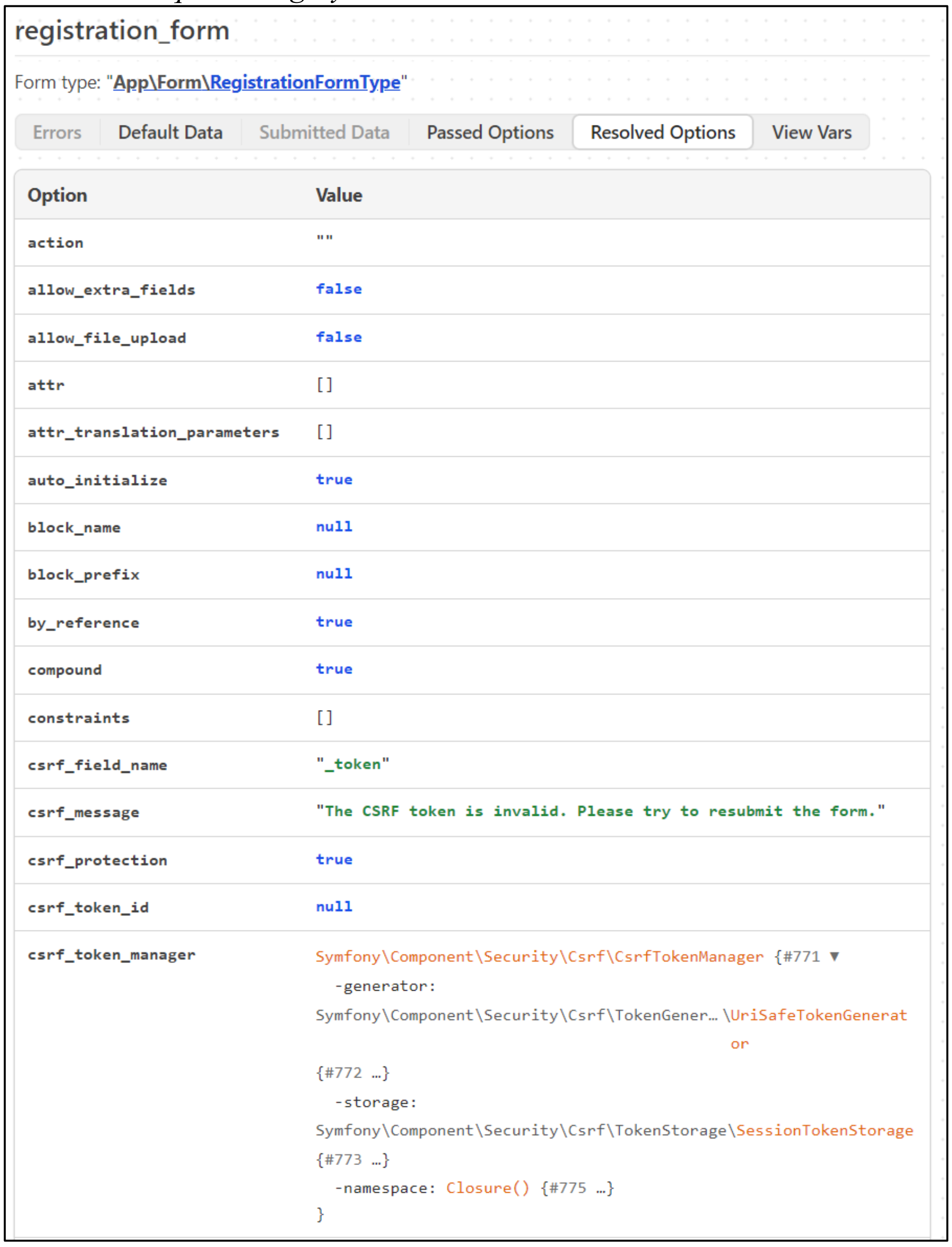
POST https://chef-quiz-ai-v12-375d56dc4eb0.herokuapp.com/register

x-www-form-urlencoded

	Key	Value	Description
✓	name	Test User	
✓	email	testuser@example.com	
✓	plainPassword[first]	testpassword	
✓	plainPassword[second]	testpassword	
	Key	Value	Description

Figure 45

Cross-Site Request Forgery Test Outcome



registration_form

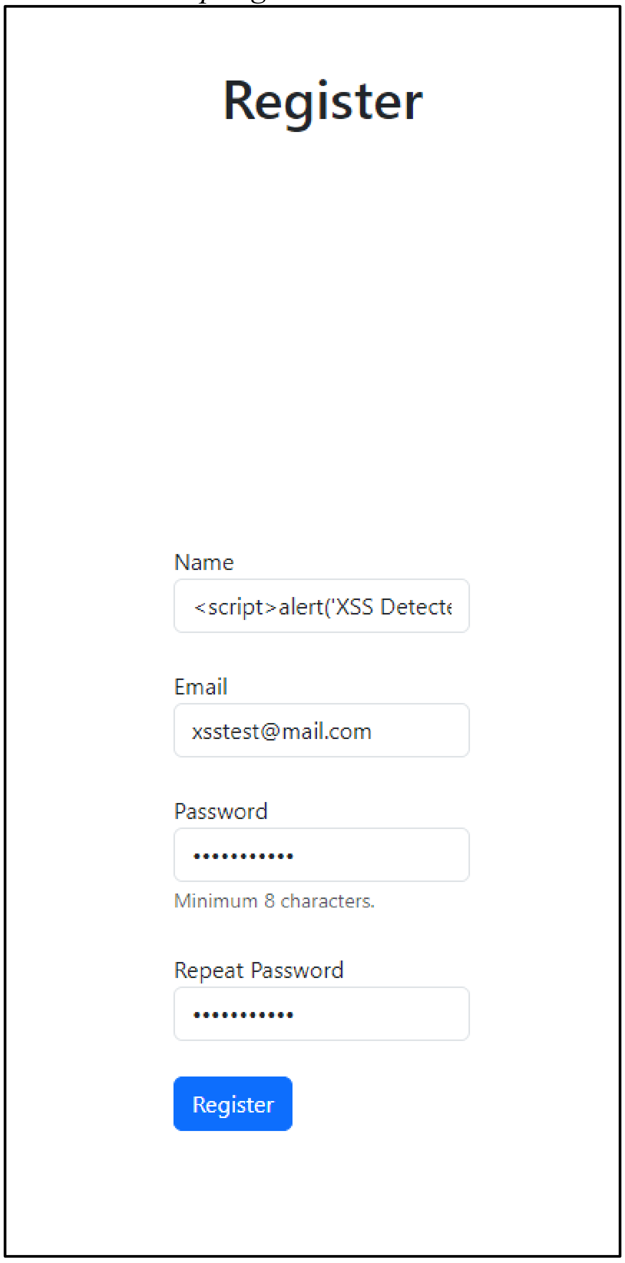
Form type: "App\Form\RegistrationFormType"

Option	Value
action	""
allow_extra_fields	false
allow_file_upload	false
attr	[]
attr_translation_parameters	[]
auto_initialize	true
block_name	null
block_prefix	null
by_reference	true
compound	true
constraints	[]
csrf_field_name	"_token"
csrf_message	"The CSRF token is invalid. Please try to resubmit the form."
csrf_protection	true
csrf_token_id	null
csrf_token_manager	Symfony\Component\Security\Csrf\CsrfTokenManager {#771 ▼ -generator: Symfony\Component\Security\Csrf\TokenGener… \UriSafeTokenGenerator {#772 …} -storage: Symfony\Component\Security\Csrf\TokenStorage\SessionTokenStorage {#773 …} -namespace: Closure() {#775 …} }

Cross-site scripting (XSS) attacks, another common threat to websites' integrity, involve an attacker injecting malicious scripts into content, expecting the script to execute in the browser when the content is served. To protect against these attacks, Symfony automatically escapes output in Twig templates so that when data is displayed, any potentially harmful characters (like <, >, "), which could be part of a script tag or JavaScript execution context, are converted to their HTML equivalents, preventing them from being interpreted by the browser as executable code. Additionally, the application uses Symfony's Form validators so that inputs must comply with constraints such as being a valid email address. To test these security measures, we used the registration page to inject, as seen in

Figure 46, a script: “<script>alert('This script executes');</script>” that generates an alert message on the screen. Upon logging in with these user credentials, the username was rendered as plain text, and the script did not execute, as demonstrated in Figure 47.

Figure 46
Cross-Site Scripting Attack



Register

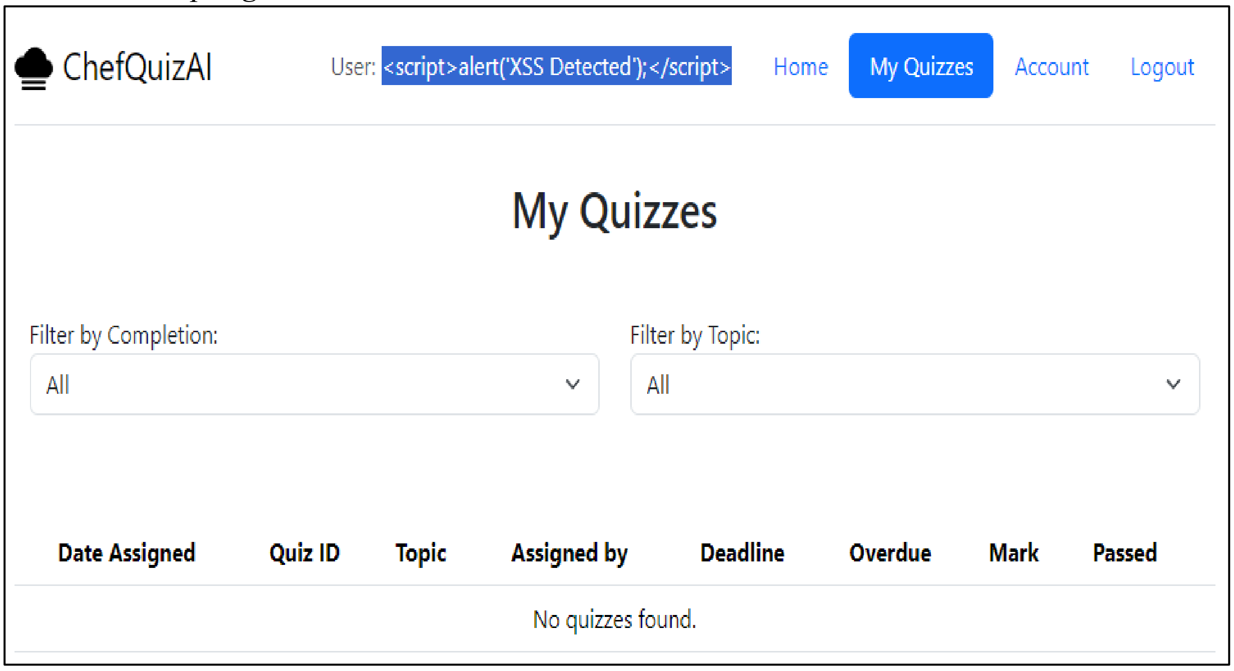
Name
<script>alert('XSS Detecte

Email
xsstest@mail.com

Password
•••••••••••
Minimum 8 characters.

Repeat Password
•••••••••••

Figure 47
Cross-Site Scripting Attack Outcome



My Quizzes

Filter by Completion: All

Filter by Topic: All

Date Assigned	Quiz ID	Topic	Assigned by	Deadline	Overdue	Mark	Passed
No quizzes found.							

5.8. User Acceptance Testing

In line with the Agile methodology, the participation of potential users and experts was valuable in providing insights and constant feedback during each sprint. An interview guide (Appendix F) was used to interact with the participants to ensure comprehensive information was collected. This information comprised usability and functionality concerns, generated quiz content quality, suggestions for improvements and an assessment of the potential impact of using the application in professional hospitality settings.

The dynamic of the tests consisted of briefing the participants on the newly implemented functionalities and allowing them to interact with the application, testing the functionalities without any further guidance. Assistance was always available, and the users' questions were annotated and discussed later. This approach aimed to obtain information about how easily the users could find the corresponding sections within the application and how intuitive it was to use the functionalities. Thanks to the application being deployed from the start of the project and accessible via the internet, most of the tests were carried out in the same places and devices the potential users would use the application after its final release. Following the interaction with the application, an informal interview was carried out to obtain feedback and insights.

The user acceptance testing within each sprint helped to action several corrections and improvements in the application, on many occasions during the same sprint and on other occasions added to the product backlog and addressed in a later sprint. These improvements included adding a delete button from the accordion of questions during the quiz generation process so a question can be deleted after it was added to the accordion but before the quiz was approved, implementing a quiz preview functionality before a quiz was approved, and implementing coloured elements to indicate pass status in the My Quizzes page. Similarly, improvements in the level of detail of generated quiz questions content and user interface features, such as filtering lists of quizzes and paginating results, were also implemented because of the received feedback. Similarly, the integration of the users during the development process allowed the identification and fixing of bugs, such as users being logged out after updating their passwords and quiz marks not being properly displayed after quiz completion. Although the core functionality of the application with respect to the originally identified requirements was largely achieved, as indicated by the overall positive response of the potential users, several features were added to the product backlog because of received feedback during the user acceptance testing on each sprint could not be actioned due to the time constraints of the project. These features include a dashboard with graphical statistics about quiz results, the possibility of generating questions with multiple correct options and timed quizzes, tooltips to provide instructions in input elements and the possibility of saving the state of a quiz during the quiz-taking process with the intention of finishing it at a later occasion.

Particular attention was given during the user acceptance process to ensure that the quality of the generated quiz questions was within the users' expectations. The assessment of this aspect of the application was initially carried out based on human evaluation and judgement. The intention was to ensure that the generated content was coherent, logical, relevant, varied and engaging. An initial reaction was that the content was not sufficiently detailed. The expectation was that the quizzes could assess objective knowledge about legislation, rules, regulations, and best practices, of which the language model was unaware and consequently did not include this level of detail in the generated questions. A further improvement of the application using prompt engineering techniques and a source of contextual information through a database containing a bespoke and relevant corpus of knowledge obtained a much higher level of acceptance. Additionally, after this approach was adopted, substantial testing was carried out to validate the factual accuracy of the generated questions, confirming the total absence of inaccuracies in the generated contents and the satisfactory level of user acceptance.

Chapter 6. Critical Evaluation

The project successfully met most of its proposed objectives (Appendix A). However, some were not fully accomplished or even attempted due to the project's time constraints and excessively optimistic expectations, mostly due to the lack of previous experience in planning projects of this magnitude. Consequently, a key takeaway from this project execution was the acquired understanding of realistic project planning. Table 31 evaluates the project achievements against the originally planned objectives.

Table 31
Review of Project Achievements Against Objectives

Main Objectives	**Achieved**
To plan a web-based application in response to specific requirements.	Yes
To develop a web-based application.	Yes
To integrate an application with an existing NLP model.	Yes
To apply AI prompt engineering techniques to optimise interactions with the GPT-4 API.	Yes
To use metrics to assess and ensure accuracy and safety in AI-generated content.	Partially
To produce a graphic user interface.	Yes
To test a software application using professional quality assurance techniques.	Partially
To deploy an application using a cloud platform.	Yes
To manage a software development project using professional project management tools and methodologies.	Yes
Optional Objectives	
To use AI techniques to provide an application with multilingual support.	Not Attempted
To use AI techniques to provide an application with image generation capabilities.	Not Attempted
To implement transfer learning techniques to train an existing large natural language model.	Not Attempted

Some initially planned approaches, such as using the Singleton design pattern to connect the application with a database, ensuring there was only one instance of connection in the session, or the Factory design pattern to generate different variants of a Quiz object, ended up being unnecessary due to alternative approaches offered by the Symfony framework. Similarly, the initial plan to implement a Retrieval Augmented Generation approach using the Elasticsearch engine to enhance quiz content accuracy was found to be problematic due to the impossibility of integrating a live retrieval system with the OpenAI API because of the lack of access to the internal model of GPT-4 Additionally, setting up and maintaining an Elasticsearch system would have required substantial setting up efforts and proved to be far more challenging than anticipated due to the project's time constraints. Given these challenges, the project shifted towards a more manageable solution that retained the principle of enhancing the prompt response using a custom food safety best practices static data repository and randomly selecting information to be passed as contextual information in the prompt.

Regarding the objective of assessing and ensuring accuracy and safety in AI-generated content using metrics, it was achieved to the extent of assessing that generated content was factually correct using human expertise and evaluation in the context of Agile methodology and comparing the content with the corpus of knowledge used in the prompt construction process. However, this assessment could have been improved using a more systematic approach, developing tailored factual assessment metrics for the food safety domain and incorporating a feedback mechanism in the same application to interact with these metrics.

Additionally, despite using professional quality assurance techniques to test the application, as demonstrated in Chapter 5, the testing strategy could have been improved even further by automating the tests and integrating several tools used in the deployment pipeline using a CI/CD approach. Furthermore, the testing approach falls short on load and performance testing under the scenario of concurrent users connecting from multiple devices, something that could be relevant in the future as the application's user base grows and the need for scalability becomes more apparent.

Despite the application not having achieved all the initial objectives, it successfully achieved its main intended purpose, namely, using an AI-powered web application to automatise and expedite the generation of assessment resources for food safety within the hospitality industry and the assessment process itself, providing a usable tool to those in charge to enhance the food safety awareness within their teams and provide continuous food safety training via the generated feedback.

Additionally, the application’s ability to produce accurate, factually correct, relevant, and detailed quiz content, achieved through prompt engineering and using a dedicated source of food safety best practices database to provide context to the language model, is a significant accomplishment. However, the dependence on an external API is a disadvantage that could be addressed by exploring alternative or concurrent solutions.

During the execution of this project, several valuable lessons were learned, including effective communication with stakeholders, problem-solving, and adaptability to solve arising difficulties. Experiencing the advantages of the Agile iterative approach was particularly advantageous, as was having the opportunity to learn several coding, testing, and deployment techniques.

In summary, the developed product largely met the expectations, as demonstrated by the functional and user acceptance testing results, and the core functionality and functional requirements were achieved. However, like any software product, it is subject to further improvements as it moves through the next steps of the software development life cycle and continues to evolve in the context of the iterative approach of the Agile methodology.

Conclusions

This project looked at implementing an AI-based website to assess food safety knowledge in restaurant teams within the ongoing food safety training that restaurants are expected to provide. The implemented solution implied the integration of the web application with a model language to provide it with generative language capabilities and the use of an integrated database containing a corpus of knowledge to provide the language model with detailed context. It maintained a human-in-the-loop approach that allowed the use of human expertise and judgement to edit and approve the generated content.

The final product demonstrated that the adopted approach successfully achieved the expected characteristics of usability and quality, relevance, level of detail and factual accuracy of the generated quiz content. The application represents a substantial improvement with respect to the state of the art in the market of food safety assessment web applications identified in the literature review. This improvement is demonstrated in the novel way this application can produce dynamic assessment resources, using case scenarios that are unique and varied, including detailed food safety-related information, and providing equally detailed feedback.

It is expected that this application will represent a significant positive impact on the levels of food safety awareness and knowledge among the users and translate into a safer dining experience for their customers, tackling the negative social impact that food poisoning and potentially fatal allergic reactions could have in the health of the wider community.

Despite the overall positive outcome of the user acceptance testing, several desired features could not be addressed due to the project's time constraints, and they remain in the product backlog, which is pending to be addressed in the future. These features include saving quiz answers for completion later, timed quizzes, multi-option for quiz answers, multiple language support, AI-generated visual elements to make the quiz content more engaging, and interactive quiz reports. Similarly, features such as a dashboard with statistical insights and improved coverage in terms of instructions and documentation could be implemented in the future.

Similarly, pending tasks include the expansion of the database used to provide context and domain-specific information to the language model, potentially exploring more sophisticated approaches of dynamically obtaining this information, such as the use of search engines like Elasticsearch, the exploration of mechanisms to avoid the dependency of the application from an external API to generate content and the application's performance for concurrent usage and heavy load perhaps exploring caching and load balancing mechanisms.

In conclusion, this project achieved its goal of enhancing food safety assessment using innovative AI-driven tools and set the groundwork for further improvements in this critical field. Moving forward, we will continue to refine the application, focusing on increasing its robustness and user-friendliness and addressing any outstanding features from the product backlog.

Bibliography

Bhoi, P. C., Singh, D., & Das, B. R. (2023). The recent advances and applications of natural language processing [Paper Presentation]. *2nd NCRTCEA.* https://www.researchgate.net/publication/370609084_The_Recent_Advances_and_Applications_of_Natural_Language_Processing

Demirag, A., Demirkol Ozturk, E., & Unal, C. (2023). Analysis and comparison of Waterfall Model and Agile approach. *Academic Journal of Information Technology, 14*(54), 181-202. https://dx.doi.org/105824/ajite.2023.03.002.x

Elmasri, R., & Navathe, S. B. (2017). *Fundamentals of Database Systems* (7th ed.). Pearson.

European Parliament and Council of the European Union. (2004). Regulation (EC) No 852/2004 of the European Parliament and of the Council of 29 April 2004 on the hygiene of foodstuffs. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32004R0852

Food Standards Agency. (2021). *Safer Food, Better Business (SFBB).* https://www.food.gov.uk/business-guidance/safer-food-better-business-sfbb

Food Standards Agency. (2021). The Food Standards Agency 2021 Food Law Practice Guide (England). https://www.food.gov.uk/about-us/food-and-feed-codes-of-practice

Food Standards Agency. (2023). *Food Allergy and Intolerance Training*. https://allergytraining.food.gov.uk/

Foster, D. (2019). *Generative Deep Learning.* O'Reilly.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Dessign Patterns.* Addison Wesley.

Great Britain. (1990). The Food Safety Act of 1990. https://www.legislation.gov.uk/ukpga/1990/16/contents

Great Britain. (1999). The Food Standards Act 1999. Retrieved from https://www.legislation.gov.uk/ukpga/1999/28/contents

Great Britain. (2013). The Food Safety and Hygiene (England) Regulations 2013. https://www.legislation.gov.uk/uksi/2013/2996/contents/made

Grønsund, T., & Aanestad, M. (2020). Augmenting the algorithm: Emerging human-in-the-loop work configurations. *The Journal of Strategic Information Systems, 29*(2), 1-16. doi:https://doi.org/10.1016/j.jsis.2020.101614

International Organization for Standardization. (2018). *Food safety management systems — Requirements for any organization in the food chain. (ISO 22000:2018).* https://www.iso.org/standard/65464.html

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., . . . Ge, B. (2023). Summary of ChatGPT-Related research and perspective towards the future. arXiv. doi:https://doi.org/10.1016/j.metrad.2023.100017

Mitchell, T. M. (1997). *MachineLearning.* McGraw-Hill.

Mizrahi, E. (2023). *Unlocking the Secrets of Prompt Engineering.* Packt Publishing.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2023). A comprehensive overview of large language models. arXiv. doi:https://arxiv.org/abs/2307.06435v6

OpenAI. (2023). *Fine-tuning*. https://platform.openai.com/docs/guides/fine-tuning

Ram, O., Levine, Y., Dalmedigos , I., Muhkgay, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023). In-Context Retrieval-Augmented language models. arXiv. doi:https://doi.org/10.48550/arXiv.2302.00083

Ronanki, K., Cabrero-Daniel, B., Horkoff, J., & Berger, C. (2023). Requirements engineering using Generative AI: Prompts and prompting patterns. arXiv. doi:https://doi.org/10.48550/arXiv.2311.03832

Russell, S., & Norvig, P. (2022). *Artificial Intelligence - A Modern Approach* (4th. ed.). Pearson.

Sommerville, I. (2016). *Software Engineering* (Vol. 10th). Pearson.

Song, F., Sun, J., & Wang, T. (2019). Overview of natural language processing technologies and rationales in application. *Theory and Practice in Language Studies, 10*(1), 49-54. doi:http://dx.doi.org/10.17507/tpls.1001.07

U.S. Food & Drug Administration. (2022). *HACCP Principles & Application Guidelines*. https://www.fda.gov/food/hazard-analysis-critical-control-point-haccp/haccp-principles-application-guidelines