2024

Moral Approach/Avoidance by the Self Task (MAAT): An Interactive Web Application

CSC 400 SUMMER 2024 CAPSTONE PROJECT

NICHOLAS OSWALD AND JOEY HILLS

SOUTHERN CONNECTICUT STATE UNIVERSITY | COMPUTER SCIENCE

Contents

1. Introduction

This report documents the development phases of our capstone project for the course CSC 400. The project, titled "Moral Approach/Avoidance by the Self Task: An Interactive Web Application," was developed by Nicholas Oswald and Joey Hills. The project was undertaken at the request of a graduate student, Paul McKee, from Duke University. Paul needed an efficient and user-friendly platform to conduct psychological experiments online.

Our web application addresses the need for a comprehensive, customizable platform that simplifies the process of creating, managing, and conducting psychological experiments. The application is designed to meet the requirements of researchers like Paul McKee and has the potential to be used by other researchers in the psychological field.

Working with a client to achieve an adequate product comes with inherent challenges. Our collaboration involved understanding his specific research needs and translating them into functional requirements for the application.

Client-Based Motivation

Paul's research required a platform that supported customizable experiments with specific parameters. Existing tools either lacked these features or were too complex to use. Our goal was to create a streamlined, intuitive platform that allowed researchers to design their experiments easily, manage participants, and collect data efficiently.

Challenges and Solutions

Throughout the project, we encountered several challenges that required innovative solutions:

Understanding Client Requirements

- **Challenge**: Initially, translating Paul's specialized research needs into technical requirements was complex.
- **Solution**: We held multiple meetings and maintained continuous communication with Paul to iteratively refine the requirements and ensure that the application met his needs accurately.

Experiment Customization

- **Challenge**: The application needed to support highly customizable experiment parameters.
- **Solution**: We implemented a flexible configuration module allowing researchers to adjust text size, colors, trial configurations, and upload images for backgrounds. This flexibility ensured the platform could accommodate a wide range of experimental designs.

Real-Time Data Collection and Monitoring

- **Challenge**: Implementing a system that could collect and monitor data in real-time posed significant technical difficulties.
- **Solution**: We utilized efficient algorithms to log data dynamically and provide live updates on experiment progress. For example, we implemented an algorithm to randomize trial presentations while maintaining balanced conditions, meeting the research requirements.

User-Friendly Design

- **Challenge**: Ensuring the application was intuitive and easy to use for researchers and participants.
- **Solution**: We employed user-centered design principles, conducted usability testing, and incorporated feedback to improve the interface. The application was built with responsive design using Bootstrap, ensuring compatibility across various devices.

Project Potential

This project not only meets the requirements of our client Paul McKee but also has broader potential applications. The platform is designed to be scalable and adaptable, making it suitable for other researchers in the field of psychology. By providing a robust and versatile tool, we aim to support a wide range of experimental designs and facilitate better research outcomes.

In this report, we will document the various phases of our project, from inception to implementation and testing. We will revise the initial System Requirements Specification (SRS) to reflect the final realization of the project and provide detailed documentation of the implementation, core algorithms, database design, external APIs/libraries, and user/installation guides. We will also evaluate the project's state, discuss testing and evaluation methods, reflect on the lessons learned, and propose future work. Finally, we will highlight the professional development and lifelong learning aspects we experienced throughout this project.

2. System Requirements Specification (SRS)

2.1. Introduction

The SRS outlines the specifications and requirements for the development of the interactive web application. It has been revised to reflect the final realization of the project.

2.2. Overall Description

The application aims to simplify creating and managing psychological experiments online. It caters to researchers needing customizable experiments, efficient participant management, and automated data collection.

2.3. Functional Requirements

- **User Registration and Authentication**: Secure login for researchers and participants with role-based access control.
- **Experiment Creation**: Customizable experiment parameters including text size, colors, and trial configurations.
- **Participant Management**: Add, edit, and delete participant details. Track participant progress and responses.
- **Data Collection**: Automatic logging of participant responses, reaction times, and accuracy. Export options (CSV, in our case).

2.4. Non-Functional Requirements

- **Usability**: Intuitive and responsive user interface.
- **Scalability**: Ability to handle a growing number of experiments and participants.
- **Security**: Data encryption and routine backups.
- **Performance**: Real-time data monitoring and efficient data retrieval.

2.5. Assumptions and Constraints

- **Platform**: The application will be hosted on the Digital Ocean platform.
- **Compatibility**: Must be compatible with modern web browsers and various devices (desktop, tablet, mobile).

3. System Design

3.1. High-Level Architecture

The system comprises a user interface built with Django templates and Bootstrap, a backend powered by Django, and a PostgreSQL database. The system architecture ensures smooth communication between components, scalability, and robustness.

3.2. Detailed Design

- **Front-End Design**: Built with Django templates and basic HTML/CSS for a responsive layout. JavaScript enhances interactivity.

- **Back-End Design**: Django framework handles data processing and business logic.
- **Database Design**: PostgreSQL for production, SQLite for development. Schemas include tables for experiments, participants, trials, and responses.

4. Implementation

4.1. Core Algorithms

The implementation includes several core algorithms:

Dynamic Trial Randomization:

- **Algorithm**: Ensures trials are presented in random order while maintaining balanced conditions.

```
import random

def randomize_trials(trials):
    random.shuffle(trials)
    return trials
```

- **Explanation**: This function shuffles the list of trials, ensuring each experiment run presents trials in a random order.

Real-Time Data Logging:

- **Algorithm**: Logs participant responses and reaction times dynamically.

```
start_time = time.perf_counter()

def log_response(participant_id, trial_id, response):
    response_time = time.perf_counter() - start_time
    save_to_database(participant_id, trial_id, response, response_time)
```

- **Explanation**: Captures the precise time a participant responds and logs it along with the response.

4.2. Database Design

- **Experiment Table**: Stores experiment metadata (id, name, description, settings).
- **Participant Table**: Stores participant details (id, name, user_id).
- **Trial Table**: Stores trial details (trial_id, experiment_id, stimuli, valence).
- **Response Table**: Stores responses (response_id, trial_id, participant_id, response_time, accuracy).

4.3. External APIs and Libraries

- **Django REST Framework**: For API endpoints.
- **PostgreSQL**: As the primary database for data storage.
- **Digital Ocean APIs**: For deployment and infrastructure management.

Our choice of Digital Ocean as the hosting platform for this project was driven by its intuitive ease-of-use and the convenience of having the database, server, and storage in one centralized location. Digital Ocean provides a straightforward user interface that simplifies the deployment and management of our web application. The platform's seamless integration capabilities allow us to host the PostgreSQL database, Django server, and static/media storage all within the same environment, reducing latency and improving performance. This centralized setup enhances the overall efficiency of our infrastructure, making it easier to scale and maintain. Additionally, Digital Ocean's comprehensive documentation and robust support ensure that any issues can be promptly addressed, contributing to the smooth and reliable operation of our application.

5. Screenshots and User/Installation Guide

5.1. Screenshots

Login:

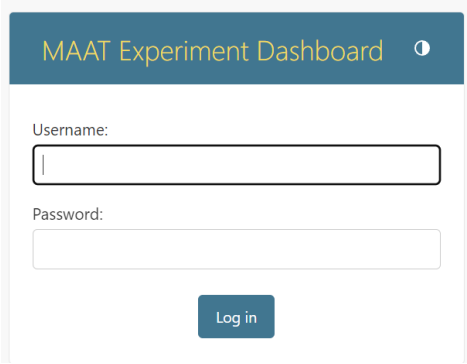

Figure 1: Login page interface.

Experiment Dashboard:

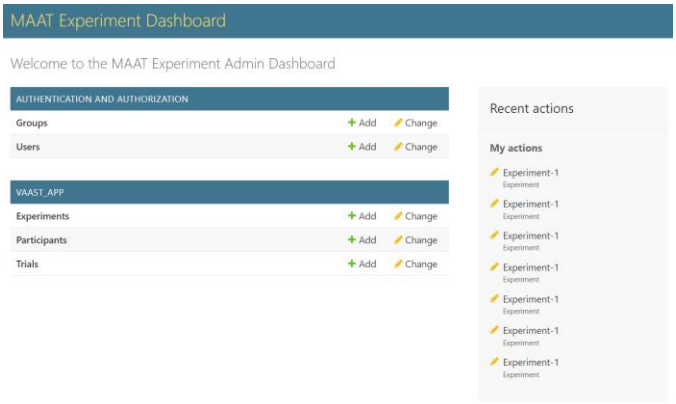

Figure 2: Experiment dashboard showcasing various experiment options.

Experiment Configuration:

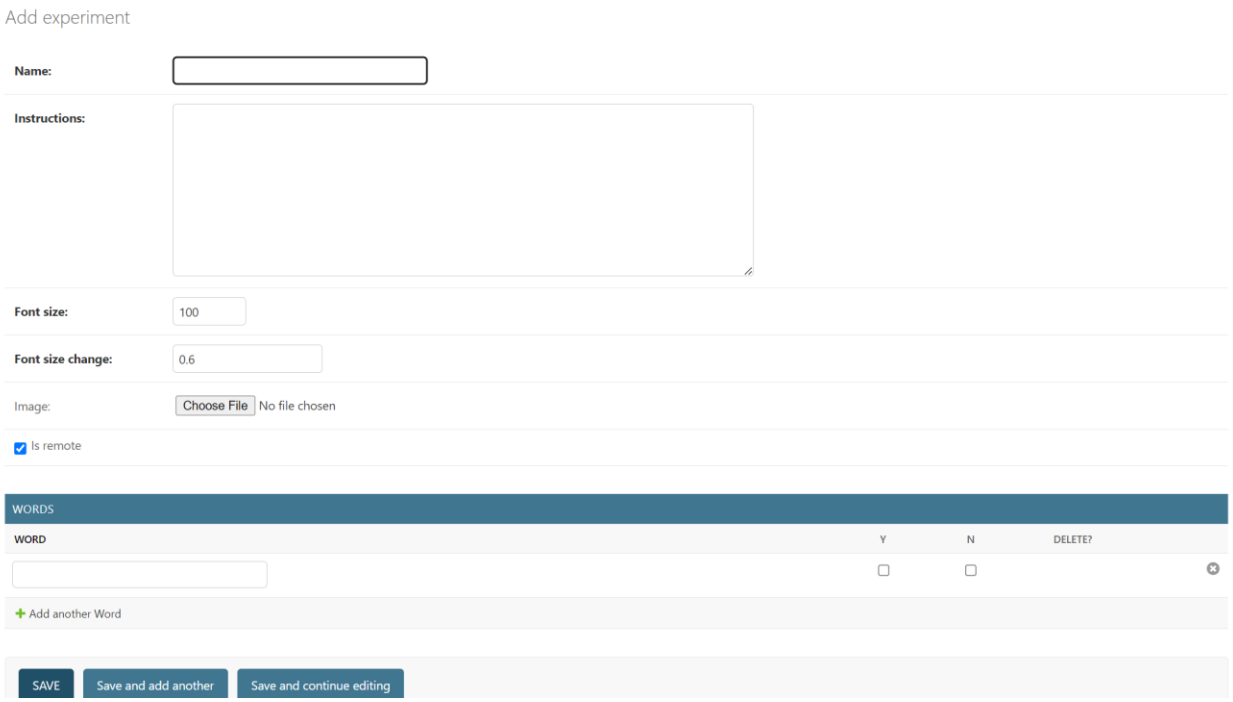

Figure 3: Interface for configuring a new experiment.

Beginning An Experiment:

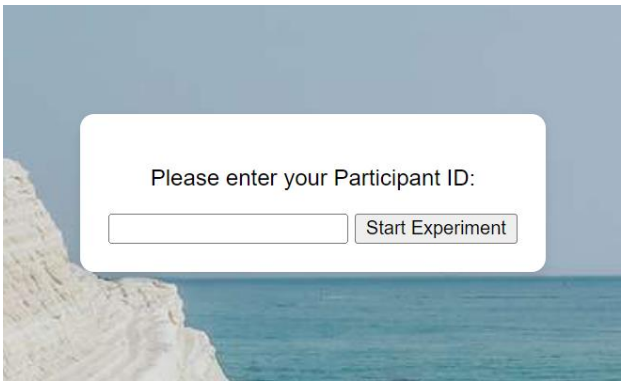

Figure 4: Participant Login

5.2. User Guide

- **Login**: Navigate to the login page, enter credentials, and select the appropriate role.
- **Create Experiment**: Access the experiment dashboard, click 'Create New Experiment', and fill out the configuration options.
- **Manage Participants**: Navigate to 'Participants', add/edit/delete participants, and assign them to experiments.
- **Run Experiment**: Select an experiment from the dashboard and initiate the experiment session.
- **View Reports**: Access the 'Reports' section to generate and view detailed experiment results.

5.3. Installation/Deployment Guide

- **Setup Environment**:

```
git clone https://github.com/noswaldSCSU/MAAT-Online.git
cd MAAT-Online
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

- **Database Configuration**:

```
# settings.py
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'yourdbname',
        'USER': 'yourdbuser',
        'PASSWORD': 'yourdbpassword',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

- **Run Migrations**:

```
python manage.py makemigrations
python manage.py migrate
```

- **Start Server**:

```
python manage.py runserver
```

- **Digital Ocean Deployment**:
    - Use the Digital Ocean API to create and configure droplets.
    - Deploy the Django application ensuring the PostgreSQL database is correctly set up on the same platform.

6. State of the Implementation

- **User Registration and Authentication**: Complete
- **Experiment Creation**: Complete
- **Participant Management**: Complete
- **Data Collection**: Complete
- **Reporting and Analytics**: Complete
- **Secure Data Storage**: Complete
- **Real-Time Data Monitoring**: In Progress
- **User-Friendly UI:** In Progress
- **Advanced Reporting and Analytics**: In Progress
- **Enhanced Security Measures**: Planned

7. Testing and Evaluation

7.1 Testing Methods:

To ensure the robustness and functionality of our application, we employed a comprehensive testing strategy comprising multiple methodologies.

Unit Testing: We began with unit testing, focusing on the smallest testable parts of our application. Each function and method was tested individually to ensure they performed correctly in isolation. This involved writing test cases for key functionalities like user authentication, experiment creation, participant management, and data logging.

Integration Testing: Once the individual components were verified, we moved on to integration testing to ensure that different modules and components interacted correctly. This included tests for APIs, database connections, and the integration of front-end and back-end components. We paid particular attention to the seamless transition of data between the front-end user interface and the back-end server and database.

User Testing: Conducting beta testing with actual researchers, including our client Paul McKee, was a crucial part of our testing strategy. We gathered feedback on usability, interface design, and functionality. This testing phase allowed us to identify and address issues that were not apparent in the earlier testing stages. We also incorporated usability testing to observe how end-users interacted with the platform in real scenarios.

Performance Testing: To ensure the application could handle the anticipated user load and data processing demands, we conducted performance testing. We simulated various usage scenarios to evaluate the system's responsiveness, stability, and scalability under different conditions. This involved stress testing, load testing, and endurance testing to identify potential performance bottlenecks.

7.2 Outcomes:

Our testing efforts yielded several positive outcomes, validating the effectiveness of our application and identifying areas for improvement.

Functionality: The project meets most of the initial objectives set out, with core functionalities like user registration, experiment creation, participant management, and data collection verified as complete and operational. Researchers can seamlessly create and manage experiments, participants can engage in trials, and data is collected and stored accurately.

Usability: User testing provided valuable insights into the application’s usability. Researchers, including Paul McKee, reported that the platform's intuitive design and functionality significantly simplified the process of conducting psychological experiments. The mobile-friendly interface ensured accessibility across various devices, enhancing user experience.

Performance: Performance testing confirmed that the application could handle concurrent users and large data sets without significant degradation in performance. The system demonstrated reliability and efficiency, maintaining data integrity and providing real-time updates as intended.

Future Meetings: As part of our ongoing commitment to refining and improving the application, we plan to schedule regular meetings with Paul McKee. These meetings will focus on reviewing the application's performance, gathering additional feedback, and fine-tuning specifications to better meet the evolving needs of users. Continuous collaboration with Paul and other users will help us identify new features, address emerging challenges, and ensure the application remains aligned with research objectives.

7.3 Future Testing:

While our current testing phase has been extensive, we recognize the need for ongoing testing and evaluation to maintain and enhance the application.

Additional User Testing: We plan to expand our user testing to include a broader participant base. This will provide more diverse feedback and help us identify usability issues that may not have been apparent in the initial rounds of testing. Future testing phases will involve more researchers and participants to ensure the platform meets a wide range of user needs.

Performance Evaluations: Regular performance evaluations will be conducted to ensure the system continues to operate efficiently as the number of users and experiments grows. This will involve periodic stress testing and load testing to identify and address any performance bottlenecks.

Security Audits: Ensuring data security and privacy is a continuous process. We will conduct regular security audits to identify and mitigate potential vulnerabilities. This includes reviewing encryption standards, authentication mechanisms, and access controls to protect sensitive participant data.

7.4 Feature Testing:

As we implement new features and enhancements based on feedback from future meetings with Paul McKee and other users, we will conduct thorough testing to ensure these additions integrate smoothly with the existing system. This will help maintain the application's stability and reliability.

Our comprehensive testing strategy has validated the core functionalities of our application, identified areas for improvement, and set the stage for ongoing testing and development. Continuous collaboration with our client and users will ensure the application evolves to meet their needs and remains a valuable tool for conducting psychological research.

8. Lessons Learned and Reflection

Throughout the development of our project, we encountered numerous challenges and learning opportunities that have significantly shaped our understanding and skills as aspiring software

engineers. Working closely with our client, Paul McKee from Duke University, we were constantly pushed to adapt and refine our approach to ensure we delivered a product that truly met his research needs. This experience taught us invaluable lessons that will be instrumental in our future careers.

One of the most critical lessons we learned was the importance of early and continuous user feedback. Initially, translating Paul's specialized research requirements into technical functionalities was a complex task. However, by maintaining open and continuous communication, we could iteratively refine our requirements and development process. This approach ensured that our development remained aligned with Paul's expectations and that any potential misunderstandings were addressed promptly. This iterative feedback loop not only enhanced the final product's usability but also underscored the importance of user-centered design in software development.

Flexibility in design emerged as another key takeaway from the project. Our initial design had to be adaptable to accommodate the changing requirements and additional features requested during the development process. By adopting a flexible architecture, we were able to implement changes and integrate new features without significant overhauls. This adaptability not only saved time but also made the development process more efficient and responsive to feedback. It highlighted the importance of designing systems that can evolve as requirements change, a principle that is crucial for long-term project success.

The project also reinforced the value of effective version control and collaborative development practices. Early on, we faced challenges with repository management, leading to setbacks and integration issues. Through these experiences, we learned the importance of maintaining a clean and well-documented version control system. Implementing best practices for repository management, such as regular commits, clear commit messages, and structured branching strategies, ensured smoother collaboration and reduced the likelihood of integration conflicts. This practice will be invaluable in any team-based software development environment.

Working with real-time data collection and monitoring introduced us to the complexities of ensuring data accuracy and system performance. Developing algorithms to log data dynamically and provide live updates required careful consideration of performance optimization and error handling. This experience taught us to prioritize efficiency and robustness in our coding practices, ensuring that our system could handle the demands of real-time data processing while maintaining accuracy and reliability.

Deploying our application on the Digital Ocean platform was a significant learning experience that broadened our understanding of cloud infrastructure and its practical applications. Setting up and maintaining a live production environment required us to address various challenges, from configuring environments to managing server resources effectively. Through this process, we gained a deeper appreciation for the role of cloud platforms in modern software development, particularly their scalability, flexibility, and integration capabilities. Understanding how to deploy and maintain applications in a cloud environment is a crucial skill for any software engineer, and this project provided us with hands-on experience that will be directly applicable in our future careers.

Reflecting on the entire project, if given the chance to start over, we would implement a more robust unit testing framework from the outset. Although we conducted extensive testing, having a structured unit testing process earlier in the development cycle would have further enhanced the reliability of our code and made debugging more efficient. Additionally, allocating more time for user testing phases earlier in the project would have provided valuable insights and allowed us to address usability concerns proactively.

In conclusion, this project has been a defining experience in our academic journey, equipping us with essential skills and knowledge for our future careers as software engineers. The challenges we faced and overcame have reinforced the importance of flexibility, user feedback, and effective development practices. As we prepare to enter the workforce, we feel confident in our abilities to tackle new challenges, continuously learn and adapt, and contribute to innovative software solutions. We are excited to bring the lessons learned from this project into our professional lives, where we can continue to grow and make a meaningful impact in the field of software development.

9. Version 2 and Further Work

Given additional time, there are several key areas we would focus on to further enhance the functionality, security, and usability of our application. These improvements are aimed at making the platform not only more robust but also more beneficial for researchers conducting psychological studies.

Enhanced Security Measures

Security is paramount in any web application, especially one dealing with sensitive data such as participant information and research results. In a future version, we would implement **multi-factor authentication (MFA)** to add an extra layer of security during the login process. This approach typically involves requiring users to provide two or more verification factors to gain access to their accounts, such as something they know (password), something they have (a mobile device), or something they are (fingerprint).

Additionally, we would adopt advanced encryption standards to ensure data privacy and protection. This would involve using industry-standard encryption protocols like AES-256 for data at rest and TLS (Transport Layer Security) for data in transit. Encrypting the data ensures that even if unauthorized access were to occur, the data remains unintelligible and secure. Regular security audits and vulnerability assessments would also be conducted to identify and mitigate potential risks, thereby reinforcing the integrity and confidentiality of the data.

Broader Integration

To maximize the utility of our platform, we would expand its integration capabilities with other research tools and platforms. Researchers often use a variety of software for data analysis,

presentation, and storage. Integrating our application with these tools would streamline their workflow, allowing for seamless data transfer and interoperability.

For instance, we could develop APIs or plugins to integrate with popular data analysis tools like SPSS, R, and Python's pandas library. This would facilitate direct import/export of data, making it easier for researchers to conduct advanced statistical analysis. Additionally, integrating with cloud storage services such as Google Drive, Dropbox, and OneDrive would provide researchers with flexible and scalable options for storing and sharing their data.

By enabling broader integration, we enhance the application's usability and reach, allowing researchers to leverage a comprehensive suite of tools in their experimental workflows. This interoperability would not only improve efficiency but also foster more collaborative research environments, where data and insights can be easily shared and integrated across different platforms.

Advanced AI Integration

Introducing advanced artificial intelligence (AI) capabilities would significantly enhance the analytical power and personalization of our platform. One of the primary applications of AI in our context would be to provide personalized recommendations based on participant interactions. This could involve using machine learning algorithms to analyze patterns in participant behavior, response times, and accuracy, thereby offering insights that can help refine experiments and interpret results more effectively.

For instance, AI algorithms could identify which types of stimuli elicit the most significant responses from participants and suggest adjustments to the experiment setup accordingly. Predictive analytics could be used to forecast participant behavior trends, enabling researchers to make data-driven decisions about their studies.

Moreover, implementing natural language processing (NLP) could enhance the way researchers interact with the data and the platform. NLP techniques could be used to analyze open-ended participant responses, categorize feedback, and even generate automated summaries of findings.

By integrating AI, we provide researchers with powerful tools to gain deeper insights, enhance the precision of their experiments, and ultimately drive more meaningful and impactful research outcomes.

Additional Features and Enhancements

In addition to the primary improvements listed above, several other features and enhancements would be valuable for a future version of the platform. These might include:

Enhanced Reporting and Visualization:

- Develop advanced data visualization tools that allow researchers to create interactive charts, graphs, and dashboards. Incorporating tools like D3.js or Plotly could provide more detailed and customizable visualizations.

User Collaboration Tools:

- Implement features that allow multiple researchers to collaborate on the same experiment. This could include shared dashboards, real-time editing, and comment systems to facilitate teamwork.

Mobile App Development:

- Develop a companion mobile application to allow participants to engage in experiments directly from their smartphones or tablets. This would increase accessibility and extend the reach of the experiments.

Enhanced Data Export Options:

- Expand data export options to include formats like JSON, XML, and direct integration with cloud storage services to provide more flexibility in data handling.

Automated Compliance Checks:

- Implement automated checks to ensure experiments comply with ethical guidelines and data protection regulations like GDPR. This could involve prompts for informed consent and secure data handling practices.

By focusing on these enhancements, we aim to deliver a more secure, feature-rich, and user-friendly platform that meets the evolving needs of researchers in the field of psychological studies. Our commitment to continuous improvement and innovation ensures that the platform remains a valuable tool for conducting cutting-edge research.

10. Professional Development and Lifelong Learning

We have been deeply engaged in a continuous learning journey that has significantly contributed to our growth as computer science professionals. As graduating seniors preparing to enter the workforce, we have had the invaluable opportunity to develop a variety of technical skills, learn new tools, and solve complex problems, all of which have prepared us for careers as software engineers or developers.

One of the most significant aspects of this project was the extensive use of the Django framework. Although we had some preliminary experience with web development, working on this project allowed us to dive deeper into Django, exploring its capabilities for rapid and secure web application development. We learned how to leverage Django's powerful features, such as its built-in ORM for database management, robust form handling, and middleware for request processing. This experience has equipped us with the skills to develop scalable, maintainable web applications and provided a solid foundation for future work in web development.

Additionally, this project required us to master web design principles using HTML and CSS. Ensuring our application was mobile-friendly and visually appealing across various devices was crucial. Through this process, we learned to create responsive layouts by combining flexible grids, media queries, and custom styles. This not only improved the usability of our application but also

enhanced our front-end development skills, enabling us to create user interfaces that are both functional and aesthetically pleasing.

Working with PostgreSQL was another critical component of this project. Postgres is a robust and powerful database management system and integrating it with Django allowed us to develop advanced database management skills. We gained experience in designing complex database schemas, optimizing queries for performance, and ensuring data integrity. This knowledge will be invaluable as we move forward in our careers, where efficient data management is often a cornerstone of successful software projects.

Deploying our application on the Digital Ocean platform was a significant learning curve that greatly enhanced our understanding of cloud infrastructure. We learned to manage droplets, configure environments, and use Digital Ocean's API for deployment automation. The experience of setting up and maintaining a production environment highlighted the importance of scalable infrastructure and reliable cloud services. This knowledge is directly applicable to many modern software engineering roles, where cloud platforms are a standard part of the development and deployment process.

Throughout the project, we relied heavily on documentation and resources from various communities. Official documentation for Django, HTML/CSS, PostgreSQL, and Digital Ocean was crucial in guiding our development efforts. Additionally, we participated in online forums and communities, which provided valuable insights and best practices. This collaborative approach underscored the importance of leveraging community knowledge and resources, a habit we plan to carry forward into our professional careers.

The main challenges we faced during this project included integrating diverse technologies into a cohesive application, ensuring data security and privacy, and managing repository issues and live deployment. Addressing these challenges required a combination of patience, perseverance, and continuous learning. For instance, implementing secure coding practices and encryption mechanisms was essential to protect sensitive participant data. Managing repository issues taught us the importance of version control and collaborative development practices, ensuring a smooth and efficient workflow.

As we prepare to enter the workforce, we feel confident in the skills and knowledge we have acquired through this project. The ability to continuously learn and adapt is one of the most valuable lessons we have learned. We are excited to bring this mindset to our future careers as software engineers or developers, where we can continue to grow, tackle new challenges, and contribute to innovative projects. This project has been a defining experience in our academic journey, shaping us into well-rounded and capable computer scientists ready to embark on the next chapter of our professional lives.