“StyleMe”

The Stackoverflowers
Chris Eardley, Danny Halovanic, Haja Koroma, Sebastian Tran

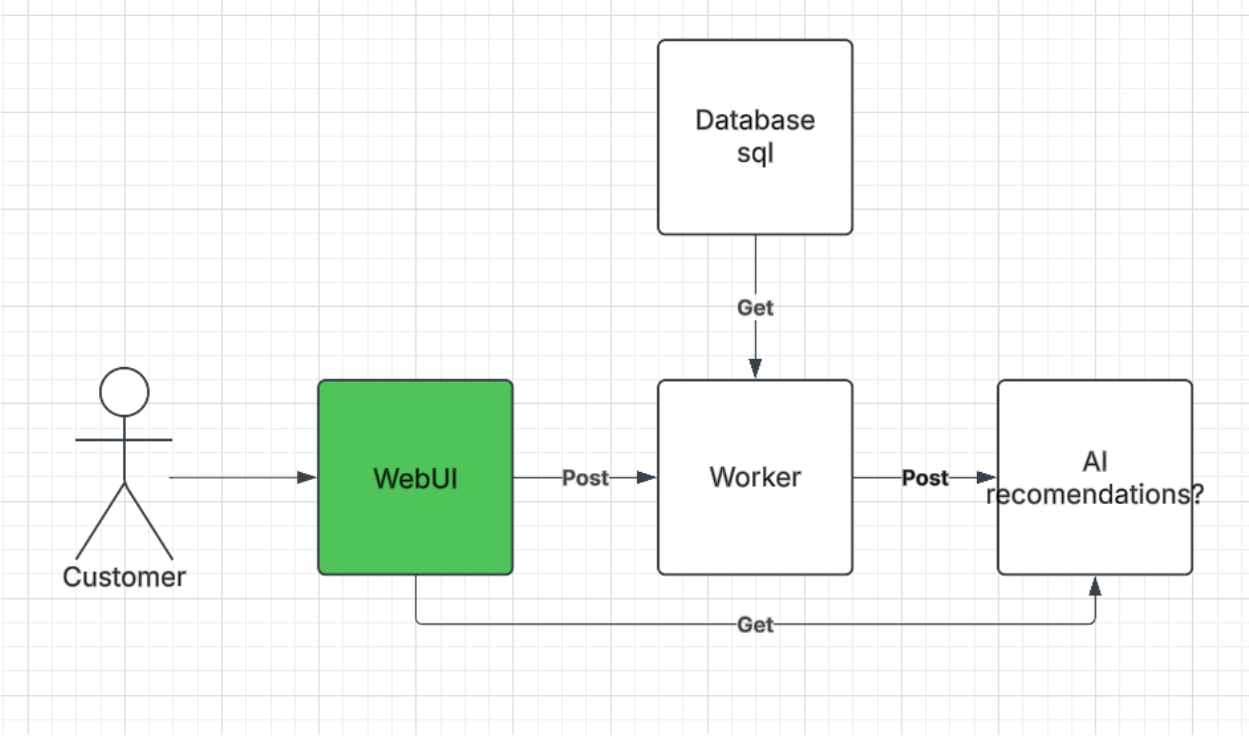
https://github.com/stran9682/468team

Chapter 1: Team Vision

Purpose of App:
Fashion and styling have been more important now than ever. With social media usage at an all time high, influencers and celebrities set trends that shape how people dress and present themselves to the public. However, finding the right outfit that perfectly compliments someone's physique and fits within the budget can be challenging.

Our Fashion App, StyleMe, is designed to help users find clothing that suits them best. Users can upload a full body photo, and our app will analyze their body shape using computer vision techniques. Based on that analysis, the app suggests clothing options that flatter their figure while staying within their specified price range.

Architecture Design:



A cloud based architecture will be used as the foundational design of our app. Our Fashion App is designed as a distributed system with several core components. The core components of our system include: WebUI, Worker, Database, and AI recommendation system.

First, the WebUI is the main interface where users upload their full body photographs and select their preferred price range. Afterwards, the WebUI sends the uploaded image along with the user preferences as a POST request to the Worker.

From there, the Worker manages the incoming data along with collecting the necessary data (e.g., past user preferences, saved styles) from the database to send to the AI recommender. It also adjusts the database to the filters the user may use. The worker essentially acts as a hub.

The Database (SQL) stores the users (gender, measurements), the products (sizes, color, price, brand).

Meanwhile, the AI Recommendation Engine analyzes the uploaded image, determines the user's body type, and provides suggestions based on physique and price range. The recommendations are then sent back to the Worker which updates the Database and forwards the response.

Lastly, the WebUI receives all of the recommendations the engine suggests and displays the details about the suggested outfits to the user.

For other design considerations, it may be possible to skip the hub all together and send all data to a large combined component that controls both the database and the recommendations. Either way, there is complexity in splitting the components as well as in implementing one large one.

Chapter 2: Implementation Proposition

In this chapter, we propose how each component in our architecture may be implemented to address the technical requirements established in chapter one. Moreover, we discuss the technologies that we plan to use in each step of the process on a per component basis.

Overall Challenges:
A challenge pinning the entire process is how to send an image from the WebUI all the way to the recommendation system. Form Files may be used, but it is critical to ensure that all components are able to accept and send them to another. Form Files are also present in most web frameworks, which hopefully will ensure compatibility across components. More broadly, enabling the docker containers to send data back and forth between each other is another challenge to address as well. Setting up the infrastructure will undoubtedly become a step in our implementation.

WebUI
Several options are available to develop and implement the WebUI with. Base Javascript and HTML is a viable option so long as it has the ability to interact with the backend components. If more utility is needed, then we can add a specific javascript library. However, most of the team does not know javascript, so some work will have to include learning.

Worker
The planned use of the worker is to bundle together the form file along with the necessary information in the database into a JSON file before sending it to the recommendation system. We can implement an ASP.NET core CORS API to fulfil the role of a hub. LINQ would make accessing and working with the database simple. Though, as suggested in chapter one, a design without a worker may also be possible. In this case, we use a direct connection to the recommendation system, and we ignore this step.

Recommendation System
For the recommendation system, Python libraries could be used. For instance, A library called OpenMMLab is used for pose estimation, which is helpful for determining body proportions. Estimating proportions could be an important part of what clothes are recommended. Another database connection would be needed for storing the clothing items with their information as part of the system.

Chapter 3: Containerization, Preliminary Testing, & Data Collection

Intro:
This chapter discusses how our team implemented containerization for our project. Containerization allows our project to be run in any environment with docker installed and promotes a modular design for our architecture. Each component in our project runs isolated inside a docker container and communicates with each other using a docker network. Variations in the implementation of this process were required in order to successfully containerize the entirety of our application. To run our project, container deployment is handled using a docker compose file.

This chapter will also discuss how clothing data was collected and how our team organizes the data in the database. Preliminary testing of our project and infrastructure will also be mentioned.

Clothing Data Collection
To provide accurate and personalized recommendations, our app, StyleMe requires a well-structured clothing dataset. We chose H&M's catalog in particular because each clothing item has a large amount of information about itself available. In order to collect this information, our team resorted to web scraping the H&M catalog. We built a custom web scraper to gather important details of a clothing item since an API was not readily available. Each clothing item is categorized based on attributes such as:

- Type (shirt, dresses, jeans, jackets)
- Fit (Loose, slim, oversized, regular)
- Color
- Price Range
- Image & Link

This data is then stored in our SQL database, which acts as the central repository for all clothing items. The database is structured using relational tables. From there, the AI Recommendation Engine queries the database to filter and suggest to best match the user's body type and Budget.

Database Organization
Our SQL database consists of these key tables

1. Users Table: Stores user details such as username, email, password and outfitsID.
2. Clothing Items Table: Contains all product details, including name, image, link, price, type, color and fit.

When structuring our data, we wanted to ensure efficient data retrieval and seamless interaction.

Preliminary Testing

Before we begin full deployment to cloudlab, our team conducted some preliminary testing on key aspects of our system

1. Database Performance Testing: We evaluated SQL queries for efficiency, checking how quickly the system retrieves clothing items based on user input.

2. Container Communication: Since our system runs on Docker containers, we tested the ability of each component that is currently running (WebUI, Worker and Database) to communicate effectively.

These initial tests are helping us identify potential bottlenecks and areas for improvement.

Web Scraper Containerization

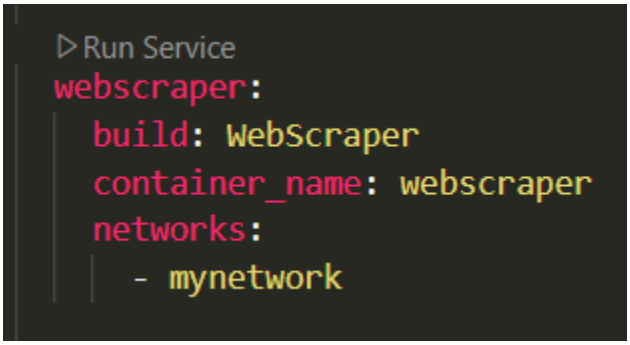
The web scraper container is initially an Ubuntu Linux image. Modifications are made to the base image to gather the necessary resources to run the web scraper. This process is primarily completed through the use of a bash script, which is copied from the local machine and run inside the container. First, the bash script installs a chrome package. Then, another necessary component, the chrome web driver, is installed and unzipped. It is then moved to the binary folder and permissions are granted so it can become executable. The final part of bash script is installing python and activating the virtual environment. Lastly, to get the web scraper source code to run, the necessary python packages are installed from a list that is copied into the container. The web scraper is connected to a docker network, and the other containers, inside the docker compose file

```
19    COPY script.sh .
20    RUN /script.sh
21    ENV PATH="/my-venv/bin:$PATH"
22
23    COPY requirements.txt .
24    RUN my-venv/bin/pip install -r requirements.txt
25
26    COPY WebScraper.py .
27    CMD ["python3", "WebScraper.py"]
```

Dockerfile for web scraper

```
apt update
apt install -y wget
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
dpkg -i google-chrome-stable_current_amd64.deb
apt-get install -f -y
wget https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/linux64/chromedriver-linux64.zip
apt install unzip
unzip chromedriver-linux64.zip
mv /chromedriver-linux64/chromedriver /bin
chown root:root /usr/bin/chromedriver
chmod +x /usr/bin/chromedriver
apt install -y python3 python3-pip python3.12-venv
python3 -m venv my-venv
. my-venv/bin/activate
```

Bash script to set up web scraper



Web Scraper inside docker compose file

Worker Containerization

Our worker component is built from base images containing the .NET 8.0 SDK and ASP.NET core runtime, providing all the tools needed to build and then run the project inside a lightweight container. The run time is the first image set up, which exposes ports 8080 and 8081. The second image, containing the SDK, builds the worker project after copying it to the container and running the necessary commands. Thereafter, another image is built off the SDK image to publish the project, which prepares the project for deployment. The last step of the dockerfile is copying the published project to the runtime container and running the “dotnet” command to start up the worker. Further setup continues inside the docker compose file. A port mapping between the container’s port 8080 and the local machine port of 8080 is created for only testing purposes. In addition, we set a dependency on the database container and add the worker to the docker network as well.

```dockerfile
# This stage is used when running from VS in fast mode (Default for Debug configuration)
FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
USER $APP_UID
WORKDIR /app
EXPOSE 8080
EXPOSE 8081


# This stage is used to build the service project
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
ARG BUILD_CONFIGURATION=Release
WORKDIR /src
COPY ["Worker.csproj", "."]
RUN dotnet restore "./Worker.csproj"
COPY . .
WORKDIR "/src/."
RUN dotnet build "./Worker.csproj" -c $BUILD_CONFIGURATION -o /app/build

# This stage is used to publish the service project to be copied to the final stage
FROM build AS publish
ARG BUILD_CONFIGURATION=Release
RUN dotnet publish "./Worker.csproj" -c $BUILD_CONFIGURATION -o /app/publish /p:UseAppHost=false

# This stage is used in production or when running from VS in regular mode (Default when not using the Debug configurati
FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "Worker.dll"]
```

Worker Dockerfile

Worker inside docker compose file

Frontend Containerization

The front end of our app uses a lightweight, node.JS alpine image, reducing space required. The process to create the frontend image is similar to the Web Scraper. First, a list of packages is copied from the local machine to the container, then a command is run to install the required dependencies. Because the frontend is set up to run on port 5173, we exposed internal port 5173 in the docker file. Specifying a specific port also serves as a reminder for which port needs CORS permitted in the worker. Lastly, once the image is built, the container and frontend is run

using “npm run dev”. The docker-compose file maps port 5173 on the local machine to port 5173 inside the container, giving the frontend access to the outside network. Publishing this port is critical because it is how the content is made available to users.

```
# Use the latest LTS version of Node.js
FROM node:18-alpine

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy application files
COPY . .

# Expose port app runs on
EXPOSE 5173

# command to run app
CMD ["npm", "run", "dev"]
```

Frontend dockerfile

```
▷ Run Service
frontend:
  build: StyleMe
  container_name: frontend
  ports:
    - "5173:5173"
  depends_on:
    - worker
  networks:
    - mynetwork
```

Frontend inside docker compose file

Database Containerization

The database did not require creating its own special image to run, but parameters were set inside the docker compose file. In the docker compose file, environment variables for the username, password, and database were added to set up the PostgreSQL database. Importantly, a persistent

volume was also made. This allows any data to continue to exist, even after the database container is shut down.

```
4       postgres:
5         image: postgres:latest
6         container_name: postgres-container
7         restart: always
8         environment:
9           POSTGRES_USER: myuser
10          POSTGRES_PASSWORD: mypassword
11          POSTGRES_DB: mydb
12        networks:
13          - mynetwork
14        volumes:
15          - postgres_data:/var/lib/postgresql/data  # Persist data
```

Creating database inside docker compose

Docker Compose

To deploy our containers as a unified system, we used a docker compose file. All the containers our project needs to run are prepared in this file. The docker compose also builds the necessary images from the dockerfiles present in our project.

Another important part of the docker compose file is creating the docker network. This allows containers to communicate with each other, but it also enables containers to communicate with each other by using a DNS rather than by IP address. For example, when the frontend needs to make a get request to the worker component for clothing data, the request can begin with, "http://worker:8080," instead of an IP address.

Chapter 4: Final Results

Challenges and Missed Milestones:

Throughout the development of StyleMe, our team encountered several technical challenges that impacted both our timeline and implementation strategy. While we made significant progress towards our overall vision, these obstacles highlighted the complexity that comes with integrating artificial intelligence, image processing and cloud technologies into a cohesive application.

Body Shape Analysis Difficulties

One of our primary challenges was accurately determining a user's body shape from a two dimensional image. Initially, we aimed to use shoulder-to-hip ratios to classify body types. However, implementing this method proved far more complex than anticipated. While our initial body shape analysis worked for people with standard physiques, our initial analysis was flawed for bodybuilders or people with a lot of muscle mass.

Limitations of Current AI Models

We also struggled to find pre-trained AI models that could analyze body shapes in a nuanced and inclusive way. While many open source solutions exist for pose estimation and landmark detection, most models lack the depth needed for detailed body classification and fashion personalization. MediaPipe, which is itself an AI driven framework, performed reliably for identifying body landmarks. However, its AI models are optimized for general pose detection, not for fine grained body type classification. Other AI models we explored either failed to differentiate effectively between body types or produced inconsistent results when tested. These limitations made it difficult to build a personalized recommendation engine based on body shape alone. As a result, we had to delay this component of the website while researching if there were any more adaptable alternatives.

Issues with Graphing Key Points

A critical step in our system's pipeline involved graphing key points on the user's image to identify landmarks like shoulders, hips and knees. After testing various tools, ultimately, we found that Google's MediaPipe provided the most reliable and consistent results for our needs. MediaPipe offered built in pose estimation and body segmentation features that allowed us to locate major joints and outlines with reasonable accuracy. While the body masks produced by MediaPipe were not perfect, it outperformed other options in terms of speed, accessibility, and ease of integration into our pipeline. The consistency of its output made it the most suitable tool for our project. We were able to extract key points effectively enough to support early versions of the body shape analysis logic.

Conclusion:

The development of StyleMe has been a valuable journey in understanding how cloud-based systems can support intelligent, user centric applications. From our initial vision of using AI to recommend fashion choices, to containerizing each component for scalability and deployment, we have seen our idea evolve into a functional system. By leveraging Docker, we ensured that each part of the application (from the frontend interface to the AI recommendation engine and database) operates smoothly within insulated containers, which enhanced our portability and maintainability. While we encountered setbacks, especially in body shape analysis and AI based recommendations, overall we gained a deeper understanding of both the limitations and potential of current AI technologies.

Although we faced challenges, these obstacles provided us with practical experience in troubleshooting real world cloud infrastructure issues. Our use of technologies such as ASP.NET, SQL, Python, and Docker Compose allowed us to not only meet most of our technical goals but also gain deeper insight into distributed system design. Additionally, our decision to rely on MediaPipe was ultimately a strong one: it proved to be the most reliable tool for verifying body visibility in user images. It gave us a solid foundation for pose estimation and opened the door for further iteration.

Moving forward, our team is committed to refining the AI recommendation engine and exploring custom training solutions that can better handle body diversity. StyleMe has the potential to evolve into a more personalized and monetizable platform. Contextual outfit suggestions could be introduced, such as recommendations for parties, business events or casual settings, to give users more relevant guidance. Affiliate linked clothing items could be integrated directly into the site, allowing users to view, click and purchase outfits easily. We also envision possibly creating a tiered experience, offering both free and premium versions of the platform.

In terms of technical growth, storing specific body parameters and maintaining a pixelated version of user uploaded images (from the neck down only) could help improve future recommendations. Additionally, the app's recommendation and interface could be refined and packaged for potential acquisition by fashion retailers such as H&M.

In the end, StyleMe stands as a promising prototype for a fashion recommendation platform. This project has strengthened our skills in full-stack development, container orchestration and cloud-native application architecture, preparing us for any future endeavors we may have in cloud computing and beyond.

Team Resumes:

Sebastian Tran

transebastian91@gmail.com - 215 284 9916 - sebastiantran.azurewebsites.net

Education

West Chester University of Pennsylvania, GPA: 4.00 *August 2023 - May 2026*

Accelerated MS + BS Computer Science, Minor in Applied Statistics West Chester, PA

Relevant Coursework

Applied Statistics, Discrete Math, Computer Systems, Foundations of CS, Data Structures and Algorithms, Intro to Cloud Computing, Software Engineering

Experience

Vans *November 2022 – August 2023*

Sales Associate Pottstown, PA

Assisted Customers in product selection through knowledge of products and communication skills.

Worked through holidays with high traffic, relying on teamwork to maintain a high level of satisfaction.

Personal Projects

Personal Website C#, ASP.NET Core, Blazor, SQLite, CSS *October 2024 - present*

- Used SQLite to store image file locations and metadata then retrieve them with LINQ asynchronously.
- Implemented interactive Blazor components with server-side rendering and component routing.
- Deployed to cloud with Azure App Service

Spam not Spam Python, MatPlotLib *September 2024 – December 2024*

- Analyzed emails in a large dataset, then categorized them using k nearest neighbors.
- Made to be adaptable to different text datasets and visualize complex data graphically using MatPlotLIb.

Rambot B C++, PROS, VEXCODE *February 2023 – April 2023*

- Lead programmer for the 2023 Spring-ford robot, Rambot B, in the 2023 Vex Robotics Spin Up competition.
- Experimented with computer vision, odometry, and optimizing PID control.
- Scored 2nd in autonomous programming win points at the state competition, securing a finalist position.

Honors And Awards

AP Scholar With Distinction *July 2023*

- Awarded to students who challenged themselves and excelled in rigorous coursework.

Create Award, Spin-Up High School Invitational at Norristown *March 2023*

- Awarded to the robotics team that developed the most unique solutions to solve game objectives.

Organizations

Secretary - Asian Student Association *September 2023 – present*

- As secretary, helped promote ASA by taking pictures of events and directly engaging with the community.
- Wrote weekly emails encouraging members to attend meetings, increasing member participation.

Skills

- **Software:** Microsoft Office: Word, PowerPoint, Excel; IBM SPSS statistics
- **Skillsets:** Object Oriented Programming, pseudocode, debugging & testing, git
- **Languages**: JavaScript & html, C, C++, Java, Python, C#, R

Christopher Eardley

608 Westwind Dr, Easton, PA 18040 ▪ (484) 903-7362 ▪ Chris.Eardley2015@gmail.com

Objective:

Seeking a summer internship in computer science to further enhance my programming and problem-solving skills. With a solid foundation in languages such as Java, C#, C++ and Python, I am eager to apply theoretical knowledge gained in the classroom to real-world projects. Dedicated to contributing innovative solutions, collaborating with diverse teams, and gaining valuable hands-on experience in software development.

Education

Easton Area High School, Easton. Pennsylvania Graduated June 2022
Relevant Coursework: Computer Science Discoveries, AP Computer Science Principles, AP Computer Science A, AP Calculus AB, Advanced Data Structures.
Activities: Easton Area Marching Band, Varsity Tennis team, Computer Science Club.
GPA: 3.70

Northampton Community College, Bethlehem, Pennsylvania. August 2022 - May 2024
Associate of Science in Computer Science
Relevant Coursework: Computer Science I & II, Calculus II, Data Structures & Algorithm Analysis, Discrete Math, Statistics, Database Systems, Computer Organization.
GPA: 3.30

West Chester University of Pennsylvania, West Chester, Pennsylvania. August 2024 - Present
Currently working towards Bachelor of Science in Computer Science
Activities: Tennis Club, Ski and Snowboard Club, Biomedical Engineering Club.
Expected Graduation – May 2026
GPA: 3.80

Work Experience

Ingersoll Rand Power Tools and Lifting; 53 Frontage Rd, Hampton, New Jersey May 2024 – August 2024
Embedded Software Engineer: Summer Internship

- Reprogrammed a Python-based Open Protocol Simulator (OPS) that facilitates communication between assembly line equipment and master controller modules, which collect, document, and analyze data from the tools. The simulator evaluates the logs generated by the controller to assist in troubleshooting the protocol.
- Configured C-Code to enable the display of multiple images on LCD modules connected to a Raspberry Pi5.
- Worked collaboratively with fellow summer interns to plan and execute an employee engagement event for the entire office.
- Engaged in various organized learning activities, including circuit board soldering, program management, standard work practices, 3D printing, and topics related to innovation and intellectual property.

Giant Food Stores; 301 Town Center Blvd, Easton, Pennsylvania September 2020 - Present
Giant Food Stores; 698 Downingtown Pike, West Chester, Pennsylvania
Store Associate: Cashier, Grocery, Gas station attendant.

Nick's Lawncare and Landscaping; Easton, Pennsylvania May 2018 – September 2021
Co-owner (brother): Maintain 20+ accounts for weekly maintenance of lawns, Landscaping, planting and mulching. equipment maintenance and repair.

Relevant Projects

Online hall-pass system for schools January 2022 - June 2022
A collaborative member of a student team tasked with developing an online hall pass system designed to track, oversee, and preserve a record of hall pass privileges, incorporating various access levels according to user roles (student, faculty, administrators). Employed Microsoft Visual Studio and C# programming for implementation.

Link List Project (to-do) October 2024 - December 2024
Using IntelliJ Integrated development environment I created a show-usage linked list in Java to prompt a user to select options (add, remove, list, next, and quit) upon startup. Coding was written using Java.

Relevant Skills

Programming: C, C++, C#, JAVA, Python.
Software: Microsoft Office, Microsoft Studio and Visual Studio code, GitHub Codespaces, IntelliJ IDEA.
LinkedIn Learning Course: Programming Concepts for Python – Completed May 12, 2024.
Communication: English (native), German (HS Language class)

Hobbies and Interests

Fitness, Tennis, Music, Snowboarding, Bowling, Disc Golf, Video Gaming, Computer and electronics construction and repair.

Haja Z. Koroma

koromahaja283@gmail.com || 267-515-2642 || Greater Philadelphia

Creative and driven Computer Science student with experience in product operations, IT support, and technical problem-solving, proficient in Java, and network activations. Successful TikTok/YouTube content creator specializing in educational content for youth, with skills in social media strategy, data analytics, and video production.

EDUCATION

B.S., Computer Science, **Minor in Digital Marketing, May 2026** || West Chester University

CS PROJECT:

- **Magic 8-Ball Simulation** https://github.com/hajak19/Eight-Ball August 2024
    - Developed a program simulating a Magic 8-Ball, utilizing Java for random answer generation.

SKILLS:

- **Languages:** Java
- **Tools:** Windows PowerShell, JGrasp, GitHub, Salesforce
- **Software:** Microsoft Excel, PowerPoint, Office, Word, VideoPad Video Editor, Adobe Photo Editor.

WORK EXPERIENCE:

College Brand Ambassador || **Cramify** || Aug. 2024 - Present || Remote

- Driving brand awareness by creating targeted TikTok and Instagram content promoting AI-powered study tools.
- Collaborating in virtual training sessions and maintaining weekly check-ins to align content with program objectives.

Intern, Technical Product Operations Specialist || **Comcast** || June 2024 - Aug. 2024 || Centennial, CO

- Identified risks and streamlined processes for installing advanced voice and ethernet services for small to medium businesses.
- Coordinated cross-functional teams to ensure the timely and accurate execution of service orders.
- Assisted with the management of complex product installations, troubleshooting, and customer satisfaction.

IT Help Assistant, **Help Desk** || **West Chester University**, Library || Aug. 2023 - Present || West Chester, PA

- Providing technical support to students and staff, troubleshooting hardware, software, and network issues both in-person and remotely.
- Streamlining issue resolution process by improving ticket response time and offering personalized support, enhancing user experience and satisfaction.

Retail Sales Associate || **CVS Pharmacy** || May 2021 - Present || Yeadon, PA

- Ensuring accurate transactions and addressing customer inquiries, enhancing satisfaction and retention through attention to detail.
- Managing inventory and restock shelves efficiently, maintaining store organization and contributing to smooth operations.

Content Creator || YoutTube || April 2019 - Present

- Leverage data analytics and trends to identify content opportunities, produce engaging and informative content, and optimize promotion—garnering 7,000 subscribers, with a focus on Gen-Z.
- Secured and facilitated 50 paid endorsement and advertising deals by building relationships with brands, negotiating terms, and creating tailored promotional content that enhances brand visibility.

AWARDS

- Phillip Fuch Computer Science Scholarship, 2024
- NCNW Bethune-Height Collegiate Career Accelerator, 2024-2025
- Reverend Anderson Porter Scholarship, 2023
- Bonnie Clarie Bruno Scholarship, 2022
- ICU Scholarship, 2022

LEADERSHIP AND PROFESSIONAL DEVELOPMENT

Rising Changemaker || **National Coucil of Negro Women** || August 2024 - Present

- Engage in mentorship with Black women professionals, participating in masterclasses and conferences focused on career development.
- Network with industry leaders, gaining access to exclusive internships and job opportunities.

Public Relations Chair || **NCNW**, West Chester University || May 2023 - Present || West Chester, PA

- Spearhead public relation campaigns and design social media strategies to elevate the National Council of Negro Women (NCNW) presence on campus, aiming to increase awareness of the organization's mission, events, and initiatives.

Member || **Computer Science Club**, West Chester University || Sep. 2022 - Present || West Chester, PA

- Organize weekly tutoring sessions for 20+ students, coordinating schedules, resources, and volunteer tutors to provide academic support in various computer science topics.

Mentor || **ICU (Intentionally Caring & United)**, Penn Wood High || Sep. 2022 - Present || Yeadon, PA

- Planned and coordinated events to implement tutoring, therapy programs and career coaching to under-represented high school students.

VOLUNTEER EXPERIENCE:

- Nile Swim Club Inauguration Ceremony Set Up and Breakdown; Christ The King Prayer Chapel Clean Up; Black Student Union Unity Day; West Chester University Move In Day 2023

INTERESTS:

- Video Editing; Content Creation; Real Estate Development; Coding; Mentoring; Volleyball; Photography; Writing

J. Daniel HALOVANIC

dan.halovanic01@gmail.com

267-648-8545

OBJECTIVE Seeking an internship opportunity that allows for practical application of my computer science and mathematical skills.

EDUCATION

Bachelor of Science in Computer Science
West Chester University, West Chester, PA
Expected Graduation: May 2025

Bachelor of Science in Applied & Computational Mathematics
West Chester University, West Chester, PA
Expected Graduation: May 2025
Cumulative GPA: 3.89

EXPERIENCE

Mathematics Tutor
Mathematics Learning Center, West Chester University
August 2023 – present

Team Member & Trainer
Chick-Fil-A, Lansdale, PA
August 2019 – present

- Train new team members in customer service, cash handling, etc.
- Assist customers with orders in English and Spanish

Maintenance and Grounds Staff
Burn Brae Day Camp of Creative Arts, Dresher PA
June 2023 – August 2023

SKILLS

R Programming
Python Programming
Mathematica Programming
Java Programming
LaTeX
Advanced proficiency in spoken and written Spanish

ACTIVITIES

The INCOMPARABLE Golden Rams Marching Band
Fall 2021 – present

Computer Science Club
Fall 2021 – present

Phi Mu Alpha Sinfonia
Spring 2022 – present

Mathematics Student Association
Fall 2023 – present

AWARDS

- Academic Excellence Scholarship
- Mathematics Honors Scholarship,
- Dean's List (Fall 2021-Spring 2024)

PROJECTS

Predictive Analysis with R
West Chester University – Fall 2023

- Developed a Model in R that was able to predict the probability of type II Diabetes in a population.
- Used R Markdown alongside ggplot2 to visualize data and write the results
- Analyzed a dataset using techniques learned in Numerical Analysis

RELEVANT COURSEWORK

- Data Structures and Algorithms
- Artificial Intelligence
- Calculus 1-3
- Numerical Analysis
- Real and Applied Analysis
- Ordinary Differential Equations