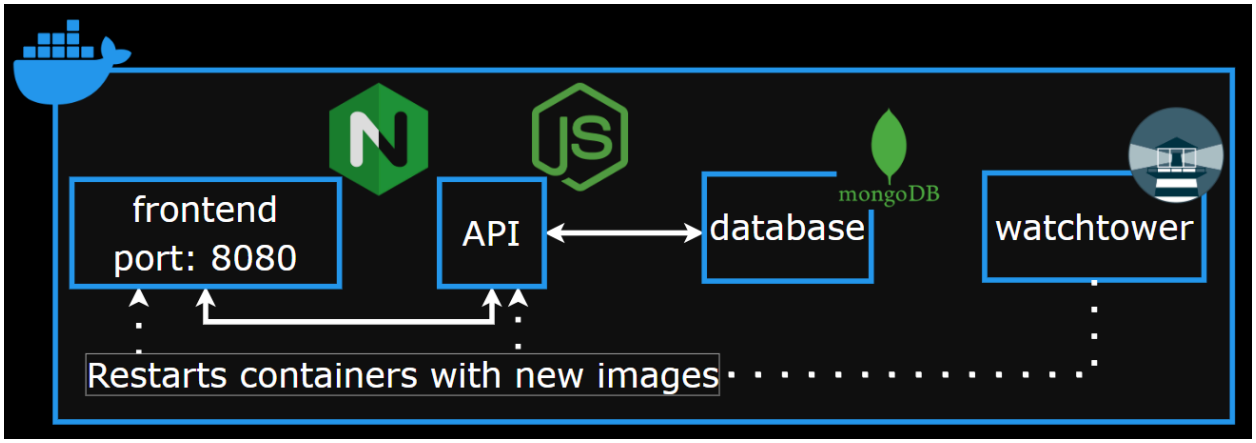
Nooh Hashmi, Demetrius Semanko, Jay Vasquez-Barreto, Evan Visalli

Chapter 1

Though the era of Telnet-accessible forums is long behind us, we at Strata Labs felt that the current browser-based forum space was either too feature-rich (Reddit), too unmoderated (4chan), or too specialized (Stack Overflow). Going forward with our project, we want to create a simple forum space that allows for posts and threads to be created, where users post replies to the post itself or to other users' post-replies. This will create a dynamic, communicative experience for our users to share messages, news, or information.

Users will be able to create an account with a username and a password in order to access the forum. Posts and replies created by the user will be stored in a database and can be accessed at any moment by the user. User authentication will be required in order to access the user's account on the forum and to view the internal forum posts to prevent crawling. Out of other forum platforms we wanted something that is simpler and moderated, an alternative to the average complicated forum page.

There are four main components in the architecture. A front-end GUI, an API service, a user authentication protocol, and a database. The front-end GUI will make calls to the user authentication protocol, and it will make calls to our API service for database read/writes; posting to the forum in all manners will be facilitated with the API service. User authentication will handle the user registration, logins, and session management. The database will store user profiles, user posts, user replies, and users' encrypted login credentials. The database can only be accessed via the API and user authentication services, but the user will be limited in which API calls they can make.



Chapter 2

We propose the following in order to implement our vision: our front-end web GUI will be built with React in conjunction with Nginx on Linux; our authentication service, built with Node.JS and Django, will process requests between the front-end and the database regarding user security, and it will be running on a Linux image; our database will be built using PostgreSQL and an official Docker image; finally, our API service, which will handle the majority of traffic- posts and replies- will be built using Node.JS and Django. All of these will be containerized Docker images, and they will be hosted on CloudLab.

Nooh will build our front-end web GUI, Demetrius will be responsible for the API service, Evan will implement our authentication service, and Jay will construct the database. We expect to have the individual components finished by Week 7, 3 March. We will have the entire project finished by Week 15, 28 April.

Chapter 3

The entire app can be launched beginning with our docker-compose.yml file. This file declares four services will run: our frontend, our API, our database, and our watchtower CI container.

Beginning with our frontend, it begins with a Dockerfile. We first pull a nginx image. Next we copy three files: index.html, forum.js, and nginx.conf. We then expose port 80. This container handles the serving of our front end HTML files to the end user so that they can view our web forum's contents (the homepage and individual threads with their replies).

Next, for our API, we first prepared a Dockerfile located in the /api directory. This image is built using a node image. We then had to 'npm install' a few packages locally on our machines before then copying package.json and package-lock.json to the image using the COPY command in the Dockerfile. After that, the image runs 'npm install' during the build process so that all dependencies are installed (note that we are NOT copying the 'node_modules' directory to the image). After this, we copy the server.js file, expose port 3000, and run the command in the container 'node server.js'.

Finally, we do not have a Dockerfile for our database, but we do use the docker-compose.yml file to streamline the containerization of the MongoDB image offered by Dockerhub. We use the docker-compose.yml file to expose port 27017 and copy our database's data to the image's database location.

Lastly, we have our watchtower container built using our docker-compose.yml file. This container is responsible for CI when the web forum is running, as when a new image is detected in the Dockerhub registry, it will kill the container running the old version of the image and then restart it with the new image that was found on the Dockerhub registry.

Final Results

We at Strata Labs have at this point of development a web forum with the following components: a front end, a back end consisting of an API and a database, and we have implemented a CI pipeline.

Static HTML files are served to the user through the NGINX container, and the user's only means of interacting with our product is via this front end container. They can view our splash page, view threads, create threads, and post replies to these threads all through our NGINX container. The NGINX container, after receiving the user's requests for either viewing or posting onto our site, reverse proxies our requests to our API container.

Our API container runs on the slim version of Node.js for faster, smaller builds, and we use Express.js for our framework inside of Node.js. The API handles GET and POST requests. Our schemas for our database are built using the mongoose middleware, and the schemas are what we use to represent our posts and threads. Each thread is its own schema where there is a subject line and a body where the user types text. Each reply to the thread is associated via the thread's object ID, so if a reply is left on a thread, the reply is associated with the thread through its object ID. Thanks to modern browsers requiring CORS to be implemented in some way, we had to require it in our API. We referred to Express.js's documentation of CORS and opted to enable all CORS requests.

For our database, we used the official MongoDB image provided to us by Docker. Configuring the container was very straightforward, and thanks to the mongoose middleware, communicating with it was simple, as our issues with POST and GET requests were because of either the API container or the front end container.

We integrated a CI pipeline using GitHub actions, the Dockerhub registry, and the watchtower image. Watchtower's purpose is to pull images from the Dockerhub registry, then to kill containers using the old image, and finally to restart that container with the new image. In order to streamline pushing the new images to the Dockerhub registry, we used GitHub's built in actions. The action runner was configured such that if a new push was detected on our main branch, GitHub would push the desired images (these desired images are configured in the publish.yml file) to the Dockerhub registry. In order to gain access to pushing to the Dockerhub registry, we provided GitHub secrets and then used those secrets in our workflow file.

In our original project plans, we had wanted to incorporate a user account feature where users would be able to post under a username, they would have a password associated with that username, and their login status would be managed with JSON web tokens. Due to the added complexity and the fact that we would be storing these passwords on a container running on CloudLab, we felt that it was both too costly in terms of time and too risky in terms of security to implement this type of user authentication.

All of these components came together to create the Strata Labs web forum known as The Lithosphere. On The Lithosphere, users can create their own forum threads. Each thread has a subject line (a title) and a body. Our users are encouraged to post about whatever they like

within reason. What they post should not be harmful or hurtful. Users can view each thread that has been made from our homepage. If they like how a thread sounds, they can click on the thread and view the body of the thread as well as its replies (if any are present). The user can then post their own replies. Adding a username to each post is something that the user must opt into by adding the username manually; the default is anonymity.

Conclusion

We found that working as a group was very difficult. Each member's unique availability meant that getting together during class or during the weekends was our best bet to work on assignments and deliverables together, but even then, meeting up during class or over the weekends happened very infrequently. With other collegiate responsibilities as well as personal responsibilities, members at times could not join us inside or outside of class, so we were put in a position of having to make decisions on their behalf or without them so that we could progress the project toward completion.

Additionally, we learned that motivation, commitment, and expertise were predictors for how much each member was able to contribute to the project. If we were not motivated, then we were not attending class, and if we were not attending class, then we were at varying levels of comfort regarding course materials (Docker specifically). Lacking expertise in the assigned tasks meant that they were not being completed, so others among us completed the tasks. We saw that we lacked communication despite having organized the necessary channels for it (exchanging phone numbers and email addresses, making a group chat over text, and making a group chat on Discord). We communicated deadlines for deliverables, quizzes, and assignments, but there was little to no communication about completing the smaller tasks for the project as a whole, so small tasks piled up and became bigger tasks which had to be tackled after a certain point.

This project is not in the state that we want it to be in. We want to see something fully fleshed out with user accounts, an AI component, proper CI, and possibly a different database. We would like to have the user accounts be either JWT or we could offload the responsibility of user authentication to Google or GitHub accounts. Another area to extend the project is with the AI component. We would love to have an AI container interacting with the forum's posts and its users such that the AI model could create threads, reply to threads, and make replies to specific users. We could have multiple AI instances with different personalities interacting with the forum such that there is artificial activity. Regarding "proper CI", we found that watchtower was useful for restarting containers with new images, but according to their repository's README, they strongly recommend that any serious deployment of CI be done with Kubernetes. We would like to implement Kubernetes as part of our CI pipeline so that our product is more like a production service rather than an in-development app. Lastly, we would like to consider switching our database from MongoDB to a RDBMS such as PostgreSQL. The only reason to do this would be for added complexity; we don't see it making a difference for our users, but we would like to experiment with implementing it.

Nooh M. Hashmi

noohhashmi02@gmail.com

Eagleville | PA | 19403 | 484-619-9683

EDUCATION

West Chester University of Pennsylvania – 700 S High St, West Chester, PA
Bachelor of Science, Computer Science Graduation: December 2025
Selected Courses: Computer Science I, Computer Science II, Computer Science II, Computer Systems, Software Engineering, Game Development, Business and Professional Speech

EXPERIENCE

UNIQLO, King of Prussia, PA August 2024 – current
Sales Associate

- Curated customer transactions at cashier including, sales, returns, exchanges, and price adjustments, while accurately handling cash, when receiving payment and giving change
- Thoroughly worked with my team members to make sure the sales floor was fully stocked and neat for the best customer experience
- Actively helped customers with problems, guiding them to items they want, and our fitting room or registers, helping customers out with self-checkout, and giving advice on how the clothing looks on the customer

AMERICAN READING COMPANY, Bluebell, PA June 2019 – August 2023
Warehouse Associate*, Assembly*

- Contributed to the mission of ARC, an organization that strives to improve entry-level English and Spanish education through curriculum, assessments and learning partnering with 5,320 schools, 1,070 districts and 50 states
- Efficiently worked with other warehouse members in getting items off shelves and packing them to get ready for the next department to ship out
- Effectively learned what to do in different departments where help was needed so everything could perform smoothly, for example, Learned how to pack a work order, and shelved books after they were received.

School Involvement

West Chester Programming Competition (WCPC)

- Competed to complete 4 different programming problems the fastest
- Completed coding questions, for example changing strings to ints, and going through and array to arrange it in a specific order

Class Experience

Game Development

- Efficiently used GODOT to create video games
- Worked with GODOT's own Programing language GD script
- Created Projects in the classes text book and for the projects improved upon them in our own unique ways
- Collaborated with 3 or 4 other members to create our own idea as a playable video game

Software Development

- Learned the basics of software development
- Implemented all techniques that were learned to complete the homework that was assigned
- Gained the ability to create HTML diagrams and create and read flowcharts
- Learned and implemented Run-Time Polymorphism, Refactoring, and how to move certain lines of code from one file to another

Computer Systems

- Created programs in C to learn how the computer operates
- Gained knowledge on the basics of how to use nano and Command Prompt
- Used Command Prompt to move, edit and create files and directories
- Learned about the different hardware of computer

Computer science I, II, III

- Sharpened my knowledge on Java
- Learned how to use Arrays, lists, queues, stacks
- Gained experience using and creating Trees
- Learned Basic and complex concepts of coding and java

Projects

- Stone Bridge Treasure Heist, Game Development Team Projects 1
- Hoard Rush, Game Development Team Project 2
- Assignments 1, 2, 3, 4, Software Engineering
    - Used the many topics talked about to complete the code given

SKILLS

Software: Microsoft Word, Excel, PowerPoint, JGrasp, Visual Studio Code, Godot
Programming Languages: Java, C-code, Gd script

Demetrius Semanko

ds1030705@wcupa.edu | github.com/DemetriusSemanko | linkedin.com/in/demetrius-semanko

Education

West Chester University — West Chester, PA
Bachelor of Science in Computer Science — August 2023 – May 2027 (Expected)

- Current GPA: 4.0
- Dean's List: Fall 2023, Spring 2024, Fall 2024
- Honors College

Scholarships & Awards

West Chester Programming Contest (WCPC), West Chester University

- 2nd among 16 contestants at the 10th WCPC — October 2024
- 1st among 36 contestants at the 9th WCPC — April 2024
- 1st among 39 contestants at the 8th WCPC — March 2024
- 2nd among 36 contestants at the 7th WCPC — November 2023

Dr. Charles A. Garber Scholarship, West Chester University

- Awarded for having high GPA as computer science major and needing financial assistance — June 2024

Academic Excellence Scholarship, West Chester University

- Awarded for academic excellence in high school — May 2023

Code Lower Merion (CodeLM), Susquehanna International Group

- 2nd among 11 schools at the 5th CodeLM — March 2019

Course Projects

CSC 240 (Computer Science III), Java — Fall 2024

- Implemented a singly linked list data structure for tracking simple parabolic motion, 180 lines
- Implemented evolutionary symbolic regression for finding line of best fit, 809 lines
- Processed the lynching data of *The Red Record* by Ida B. Wells, 831 lines

CSC 231 (Computer Systems), Java — Fall 2024

- Reverse engineered from C object file to C source code, 28 lines

CSC 142 (Computer Science II), Java — Spring 2024

- Home field advantage analyzer with a dataset of 28,689 basketball games from 2017–2018 season
- Hangman game
- Flesch reading-ease score calculator
- Rock, paper, scissors game

CSC 141 (Computer Science I), Python — Fall 2023

- Statistical analysis of an array
- Single-operation calculator

Personal Projects

Java

- Text-based Blackjack game, 374 lines — July 2024
- Card game base package, 252 lines — June 2024

Technical Skills

Languages: Java, Python, C/C++
Developer Tools: Visual Studio Code, Eclipse JDT, Code::Blocks
Operating Systems: Microsoft Windows 10/11, Linux CLI
Platforms: ServiceNow, Microsoft Azure AD
Software: BeyondTrust Remote Support
Maths: Calculus I, Introductory Statistics

Research

Plants and Pesticides — August 2019 – December 2019

Research assistant for Dr.Vishal Shah of Biology at WCU as a HS student

- Treated plants with different pesticides
- Monitored plants for insect presence

Experience

Help Desk Agent — May 2024 – Present
West Chester University — West Chester, PA

- Acted as first point of contact for help desk consultations

Lot Attendant — March 2021 – March 2022
AutoLenders — Exton, PA

- Organized, managed intake of new vehicles
- Prepared cars for exchange, customers

Shift Lead — September 2020 – March 2021
Wayback Burgers — West Chester, PA

- Received and fulfilled customer orders, requests
- Maintained premises' front- and back-end

Jhan Vazquez-Barreto

JV966821@wcupa.edu

EDUCATION

West Chester University of Pennsylvania, West Chester, PA
Bachelor Degree in Computer Science

- GPA: 3.5

Millersville University, Millersville, PA
Major: Computer Science, August 2022 – May 2023

Delaware County Community College, Media, PA
Associate degree in Liberal Arts, August 2020 –
August 2022

Course Experience

CSC 240, Computer Science III - Java. Fall 2024

- Specialized in Object-Oriented programming involving complex programs.
- Initialized a circular linked list data structure to store data for a hotel managing system.
- Implemented a text processing program that collected and formatted data from *The Red Record* by Ida B. Wells.

CSC 231, Computer Systems. Fall 2024

- Acquired skills in understanding components of CPUs, memory, storage, networking, and operating systems.
- Reversed engineered in assembly language back to C language.

CSC 220, Foundations of Computer Science. Spring 2024

- Developed algorithmic thinking in computational logic.
- Initialized complex finite-state machines.

CSC 142, Computer Science II. Spring 2024

- Implemented core logical thinking for problem solving.
- Studied primitive types and strings inside an array.
- Initialized sorting and searching in an array.
- Introduced to objects and classes.

CSC 301, Computer Security & Ethics. Fall 2024

- Developed skill in creating secure systems.
- Developed understanding of cyber attack methods and defenses.

Personal Project

Database Structure, December 2024

- Implemented a complex singly linked list data structure that allows for large amounts of data to be stored.

Technical Profile

Languages: Java, Python, C, Racket, some assembly

Software Tools: Visual Studio Code, Eclipse IDE

Operating Systems: Linux, Windows

Relevant Courses: Discrete Structures, Calculus 1, Calculus 2, Intro to Statistics

Work Experience

Giant Food Company, Kennett Square, Pa

Replenish associate, July/2021 – Present

- Replenish stock and complete tasks in a timely manner
- Specializes in leading team work
- High drive in customer experience
- Prospers in communication skills

Lowe's Companies Inc., Avondale, Pa

Seasonal Garden Associate, May/2021 – July/2021

- Provided aid towards difficult tasks involving customers, and coworkers with care
- Sufficed in one-to-one customer service
- Drove department sales
- Collaborated successfully with other store departments

Evan Visalli

Junior Undergraduate Student

Contact

Address
325 Winding Way Glenside, PA, 19038

Phone
(267) 625 5982

E-mail
21evisalli@gmail.com

Skills

Adobe Illustrator, Adobe Photoshop, GitHub, VSCD

Eclipse, Java, Visual Studio, C

Word, Excel, Google Docs, Google Sheets, Google Slides, Microsoft Office

Education

Bachelor of Science: Computer Science

West Chester University – West Chester, PA, Anticipated Graduation June 2026

- GPA: 3.06
- Transferred from Montgomery County Community College – May 2023

Work and Project Experience

Customer Service Associate

Wawa Inc, Glenside, PA, May 2022 – Present

- Exhibited key company values to efficiently perform tasks and deliver reliable and friendly customer service.
- Ensure exceptional customer service by maintaining professionalism and patience by putting the customer first.
- Demonstrate the ability to remain organized in a fast-paced environment by establishing effective time management skills.
- Maintain back-end and front-end facilities by effectively keeping products stocked and organized.

Text Processing Project

- In this project me and a group of others were tasked with creating a program which would filter out spam emails from a csv file filled with emails. The filtering algorithm created used the Euclidean Distance to calculate how similar the target email was towards the three closest already calculated emails. The target emails value will be checked against the closest 3 emails values and the target email will then be marked either spam or not spam depending on whether there are more spam or not spam emails closer to the target emails value.

Coursework

- Computer Science I, II, III
- Computer Systems, Foundations of Computer Science
- Computer Security & Ethics
- Data Structures & Algorithms
- Data Communications and Networking I
- Introduction to Cloud Computing
- Statistics 1, Intro to Statistics 2
- Calculus I
- Fundamentals of Algebra
- Introduction to Discrete Mathematics
- Business and Organizational Writing
- Digital Media

Activities

- Orchestra, Cheltenham School District, 2013-2021
- Tennis, Cheltenham School District, 2017-2021
- Ski and Snowboard Club, Cheltenham School District, 2017-2021
- Ski and Snowboard Club, West Chester University, 2024 - present

Honors and Awards

- Transfer Recognition Scholarship, West Chester University, September 2023
- The Mitchell Levin Exemplary Student Award, 2017

Hobbies

- Listening to music
- Playing Video Games
- Researching new parts for computers
- Looking into new technologies and how they function
- Cooking
- Reading books / Comics / Manga