# Marvel-Themed Community Platform - A Web Application for Comic Enthusiasts

Jared Preyer, Paul Rodriguez, Noah Mamo,
Max Knauss, Alexis Mollet

# Demo

# Tools our group used

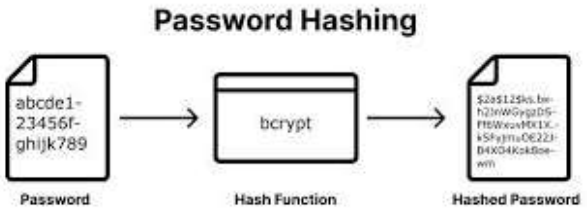
```
alexismollet@econ2-219-23-edu project % docker compose down -v
[+] Running 4/4
 ✔ Container project-web-1                Removed
 ✔ Container project-db-1                 Removed
 ✔ Volume project_CSCI-3308-group-project Removed
 ✔ Network project_default                Removed
alexismollet@econ2-219-23-edu project % docker compose up
[+] Running 4/3
 ✔ Network project_default                  Created
 ✔ Volume "project_CSCI-3308-group-project" Created
 ✔ Container project-db-1                   Created
 ✔ Container project-web-1                  Created
```

**Docker:** A platform that enables us to package our application with all its dependencies into a standardized unit for software development, ensuring consistency across various development and release cycles.

Rating: 3.5/5

**Github:** A centralized repository that allowed the team to seamlessly share code as well as using their project dashboard feature

Rating: 5/5




**bcrypt:** A powerful library for hashing passwords, employed in our application to ensure the security and integrity of user passwords.

Rating: 5/5

# Tools our group used


**PostgreSQL:** An advanced, open-source relational database, utilized for its reliability and robustness in storing and managing our application's data efficiently.

Rating: 4.5/5

```
const db = pgp(connectionString);
db.any('SELECT * FROM users WHERE id = $1', [123]);
```


**Socket.io:** A JavaScript library that enables real-time, bidirectional, and event-based communication between web clients and servers, integral for implementing features like live chat and notifications.

Rating: 5/5

**Feathers.js:** An open-source web framework for building real-time applications and REST APIs using JavaScript or TypeScript, which we used to enhance our application's real-time features, like chat.

Rating: 4/5

# Tools our group used

```
const express = require('express');
const app = express();
app.listen(3000, () => console.log('Server running on port 3000'));
```

**Express.js:** A streamlined web framework for Node.js, essential for constructing our server and API, enabling swift handling of requests and streamlined development of web and mobile applications.

Rating: 4/5

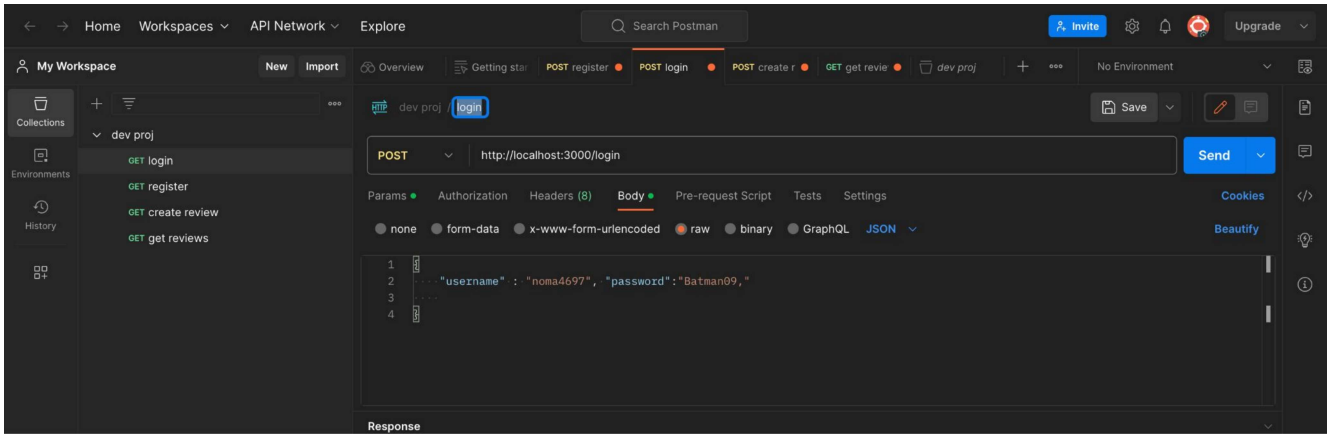
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <meta name="description" content="" />
    <!-- <h1 style="text-align: center; font-family: 'Bangers', cursive;">Marvel Comic Reviews</h1> -->

    <!-- Google Font for Comic "BOOM" Style -->
    <link href="https://fonts.googleapis.com/css2?family=Bangers&display=swap" rel="stylesheet">
    <!-- TO-DO: Add the <link> tag to include bootstrap stylesheet -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous" />


</head>

```

**EJS (Embedded JavaScript Templates):** A simple templating language that lets us generate HTML markup with plain JavaScript, used to render dynamic content on the client-side with ease and flexibility.

Rating: 5/5

# Tools our group used

**Postman:** is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.




Postman was utilized to make requests to the API routes that we wrote to check if the data was being sent in the correct format. We tested creating review and getting the reviews.

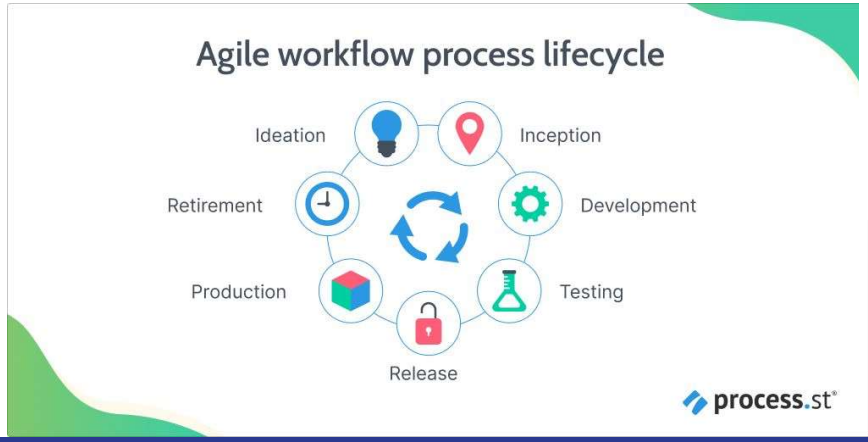
Rating: 4/5

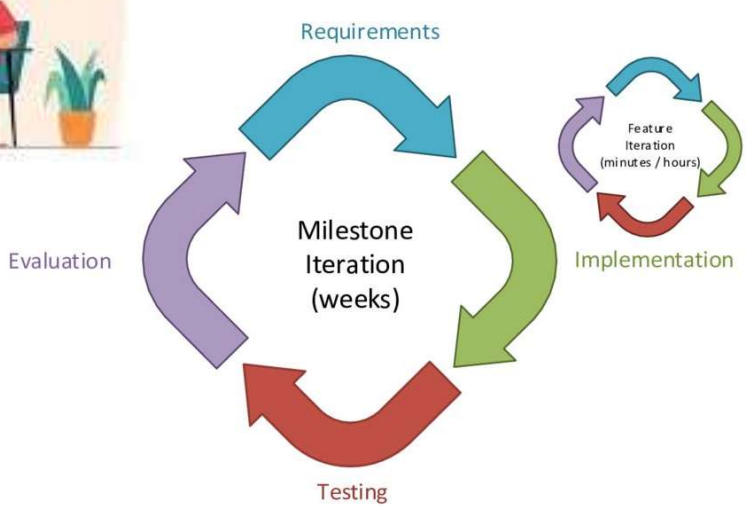
# Challenges and Pivots

- Initial API challenge and the pivot to the Marvel comic
- Rest API → Websockets for groups only
  - Authentication
- Docker issues
- Git issues

# Methodologies

- Peer Code Reviews
- Pair Programming
- Iterative
- Agile

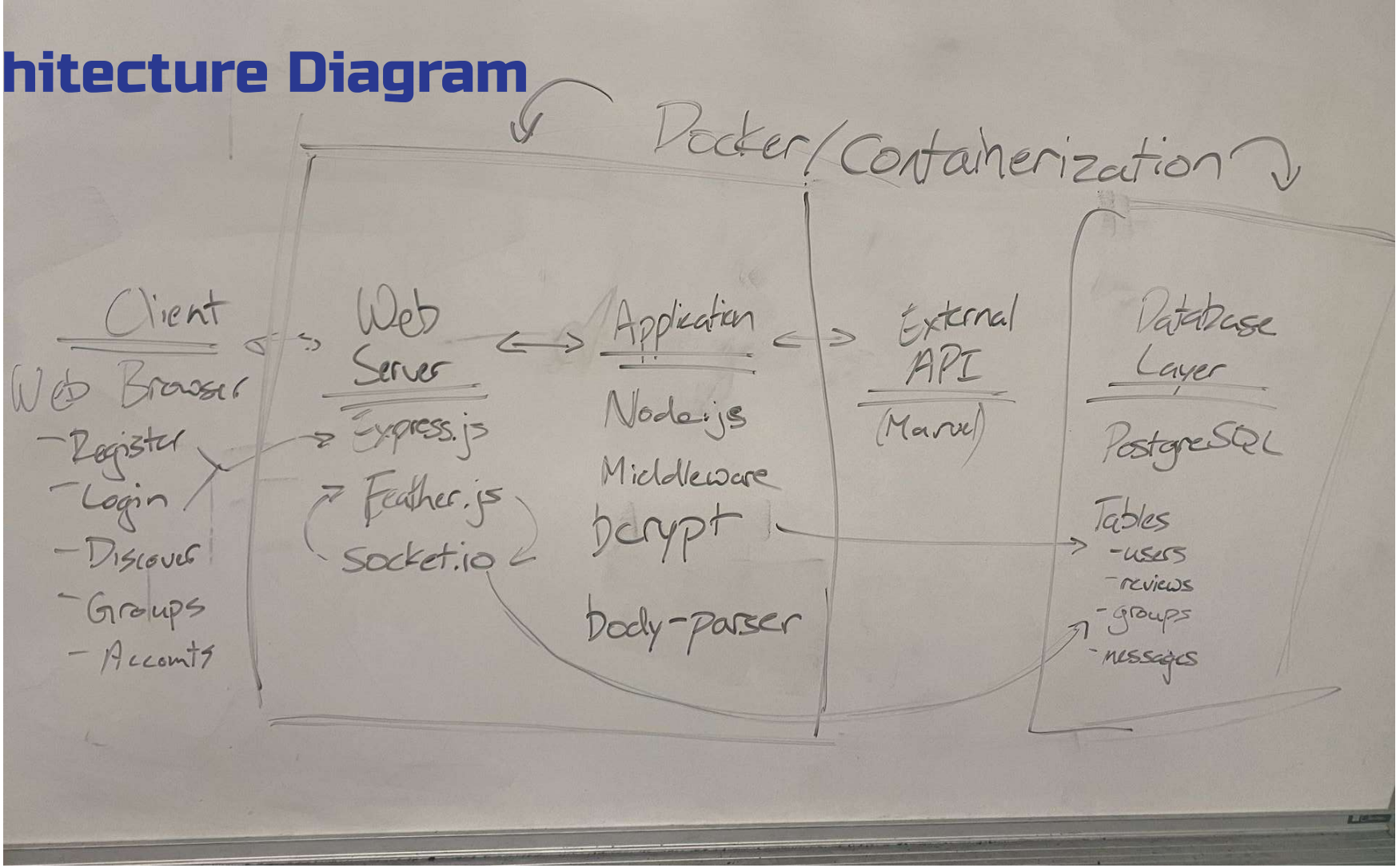


# Team Collaboration

- The vision for our group was to have the most availability to the team as possible.
- Paul handled scraping the API, setting up the databases, and maintaining discover.
- Jared handled building the group functionality,  databases, and insert statements.
- Alexis handled the account feature, testing, and the outlook of the website.
- Max was responsible for the id comic search, assisted with the feed, and implemented message functionality.
- Noah led the feed feature, implemented databases, handled the search engine.
- Overall, the team was organized and maintained the schedule that was discussed after each meeting and setting realistic goals to accomplish.

# Architecture Diagram



# Security

- Authentication
  - Session based authentication
  - Users had to logged in to access most routes
  - username/password validation: hashing, taken usernames, password reqs
- Authorization
  - Delete
    - Reviews
    - Messages
  - Both server and client side
  - Web sockets included

# Future Enhancements

- Change the site to only using web sockets, to have a social media-esque feel with multiple interactions coming in real time
- Create our own Marvel API. The current one is slower than we would like,  and its parameters are poor, such as having no full text search

# Summary -> Q&A

- Description of the project
- Tools we used
- Methodologies
- Architecture Diagram of the application
- Challenges
- Team Collaboration
- Security
- Future Enhancements
- Demo your project