

## **Curre: Shoe Recommendation Software**

### **Team Troi**

ENSE 374

Mekli Alemu, Shema Dabiri, Dmytro Stepaniuk

Dr. Timothy Maciag

Dec 2, 2022

## **Table of Contents**

<b>I) Introduction</b>	<b>2</b>
<b>II) Project Management</b>	<b>5</b>
<b>III) Prototyping</b>	<b>6</b>
<b>IV) Project Development</b>	<b>7</b>
<b>V) Programming Principles</b>	<b>8</b>
<b>VI) Conclusions and Reflections</b>	<b>9</b>

## I) Introduction

We are team Troi and our group's project is developing a running shoe recommendation software--with the intention of our North Star customer being novice runners. We finalized this idea after distilling our personal interests and having a goal to provide a solution to a real world problem.

To finalize our main idea, each team member's proposal was ranked by difficulty of implementation, understandability, and usefulness. We acknowledged there were time, technological, and financial constraints involved with the design project. Therefore, to prevent these constraints from deterring our creativity, we narrowed our north star customer and scope to complement the level of difficulty for each proposal.

Shae									
Design	Dimensions Comparison	Recommend Shoes	Recepie Filter	Spill/mess Cleanup	Bubble Tea Recommendation	Travel / Flight Destination	Professionalize a Phrase	Colorize Image	Furniture
Criteria									
Easy-to-use + understandable	10	10	10	10	7	7	7	10	7
Easy to implement	4	8	7	7	10	4	4	4	3
Usefulness	10	6	6	10	3	7	10	4	7
Meklit									
Design	Dimensions Comparison	Recommend Shoes	Recepie Filter	Spill/mess Cleanup	Bubble Tea Recommendation	Travel / Flight Destination	Professionalize a Phrase	Colorize Image	Furniture
Criteria									
Easy-to-use + understandable	9	10	10	9	10	7	6	10	7
Easy to implement	5	6	7	7	10	3	5	5	2
Usefulness	10	7	5	7	2	9	6	5	7
Dmytro									
Design	Dimensions Comparison	Recommend Shoes	Recepie Filter	Spill/mess Cleanup	Bubble Tea Recommendation	Travel / Flight Destination	Professionalize a Phrase	Colorize Image	Furniture
Criteria									
Easy-to-use + understandable	9	10	10	10	10	6	6	9	9
Easy to implement	4	8	7	7	8	3	1	3	6
Usefulness	6	4	6	7	9	8	7	4	6
Total									
Design	Dimensions Comparison	Recommend Shoes	Recepie Filter	Spill/mess Cleanup	Bubble Tea Recommendation	Travel / Flight Destination	Professionalize a Phrase	Colorize Image	Furniture
Criteria									
Easy-to-use + understandable	28	30	30	29	27	20	19	29	23
Easy to implement	13	22	21	21	28	10	10	12	11
Usefulness	26	17	17	24	14	24	23	13	20
Total	67	69	68	74	69	54	52	54	54

**Figure 1:** Pugh's method used as a decision making matrix

We highlighted our top three highest ranking ideas and discussed which we were most passionate about. We settled on creating a running shoe recommendation software because,

contrary to common misconception, there are various considerations to be made regarding the preparation and selection process for running shoes. The differing attributes of certain shoes determine the comfort, safety, and performance quality of a runner. However, the assessment process can get complicated and overwhelming for novice runners selecting their running shoes. This project's mission is to lower the entry threshold for new runners interested in incorporating running for health, competitive, or rehabilitative purposes. Overall, the application's strategic focus areas are lifestyle, education, and sport. We seek to break down the intimidating barriers to running by providing the means to accessible knowledge, educating runners on various shoe attributes, and providing a customized footwear assessment system. This was our "why" determining our motivations for the creation of this software.

Our application was implemented as a web application utilizing the following tech stack: Javascript, HTML, and CSS on the Node.js and Express.js frameworks. Additional functionality regarding the need of a shoe database prompted us to host our MongoDB database remotely on Atlas. Through these technologies, we created a website that not only had the function of a shoe recommendation software, but also served as an information hub; it distilled and presented users with more digestible information about running shoes. Recognizing the significance of user interfaces for the user experience, we settled on a low threshold and high ceiling minimalistic design that was easily understood by any user and also informative.

Our application would welcome a user, prompting them to sign in with an account after which they would go through a series of seven decision pages choosing different attributes for their shoes. The first question is about terrains--to establish the environment the users would be running in. The second question is about cushioning and the options are maximum, moderate, minimal, or barefoot. The next question asks the users if they want their shoe to possess a heel-drop or not. Then the user is provided with a choice of pronation types such as overpronation, neutral, and supination. Then the user is prompted to enter the width of the shoes whether they are wide, regular, or narrow. The last question is to find the user's comfortable spending range after which they will be redirected to their results. On the results page, users can see which shoes would fit best based on their given requirements.

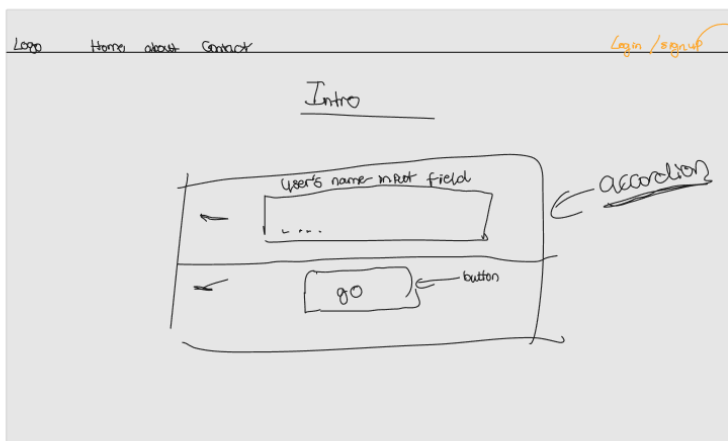
Through this application, we have reduced inaccessibility of finding appropriate running by lowering the high learning threshold. Moreover, there is now ease of access and convenience to find the necessary resources for users to educate themselves. Currently, we have improved the means of informing users about shoe requirements by providing a practical tool to access said shoes.

## **II) Project Management**

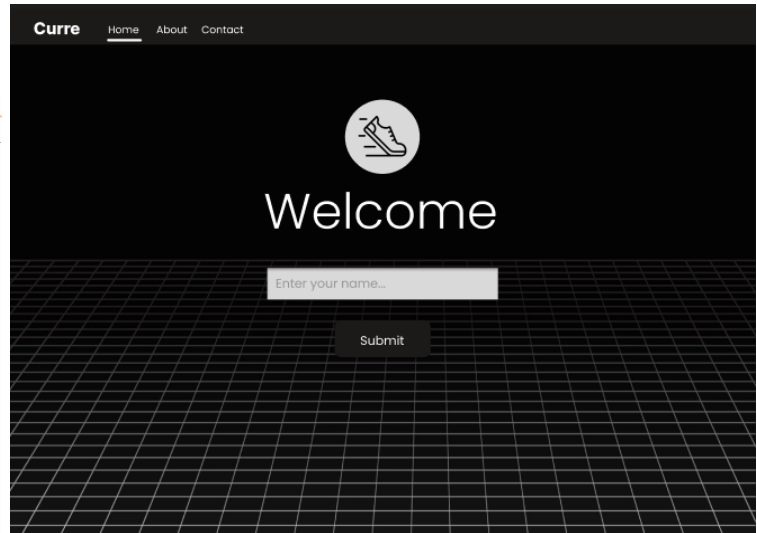
We had a formal project initiation outlining the technicalities involved in establishing and executing a project from the managerial stage. The business case document outlined the various options and benefits regarding our project idea implementation. The project charter document highlights the associated project objectives, milestones, and risks. For instance, we anticipated difficulty during the implementation of specific design features. To ensure a failsafe software design, we proposed varying, flexible designs of the MVC architecture. Furthermore, the project requirements showcase the initial minimum viable product design. The website must enable users to select shoe options and receive an appropriate recommendation. We later formally declared our project roles and responsibilities in our official document. Our project scope document states the planned deliverables. It clarifies misconceptions about the website by stating it is not a shoe purchase site or an advertisement site. Through the stakeholder engagement plan, we establish methods of engaging users, supervisor Dr. Maciag, and ourselves in our development process. We also documented each stakeholder's level of power, interest, and support with regards to its development.

## **III) Prototyping**

The first minimum viable product initially had a broad overview of the features and functionality required to meet users' needs effectively. Throughout the development process, unnecessary features were dismissed, so we could focus on making a functional tool. Enhancement and extra feature implementation followed once we addressed the user's main needs. The key functions we decided upon are as follows: the user must have the ability to enter their name and create separation from other users, go through separate pages for the shoe options, and have access to an information and contact page. The user account functionality involves deleting every session and a save function for their resulting shoes.



**Figure 2: Lo-fi diagram**

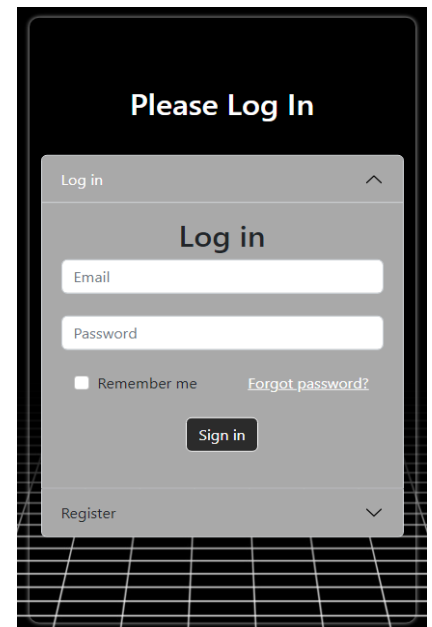


**Figure 3 & 4: Hi-fi diagrams**

Following the feedback from our scrums, we included a sign-in and register functionality and updated our result page. This user account was included in the original functionality, but we were unsure if such a function could be implemented with the tech stack we were learning. We had yet to cover user authentication in the lab tutorials at the time of our prototyping.

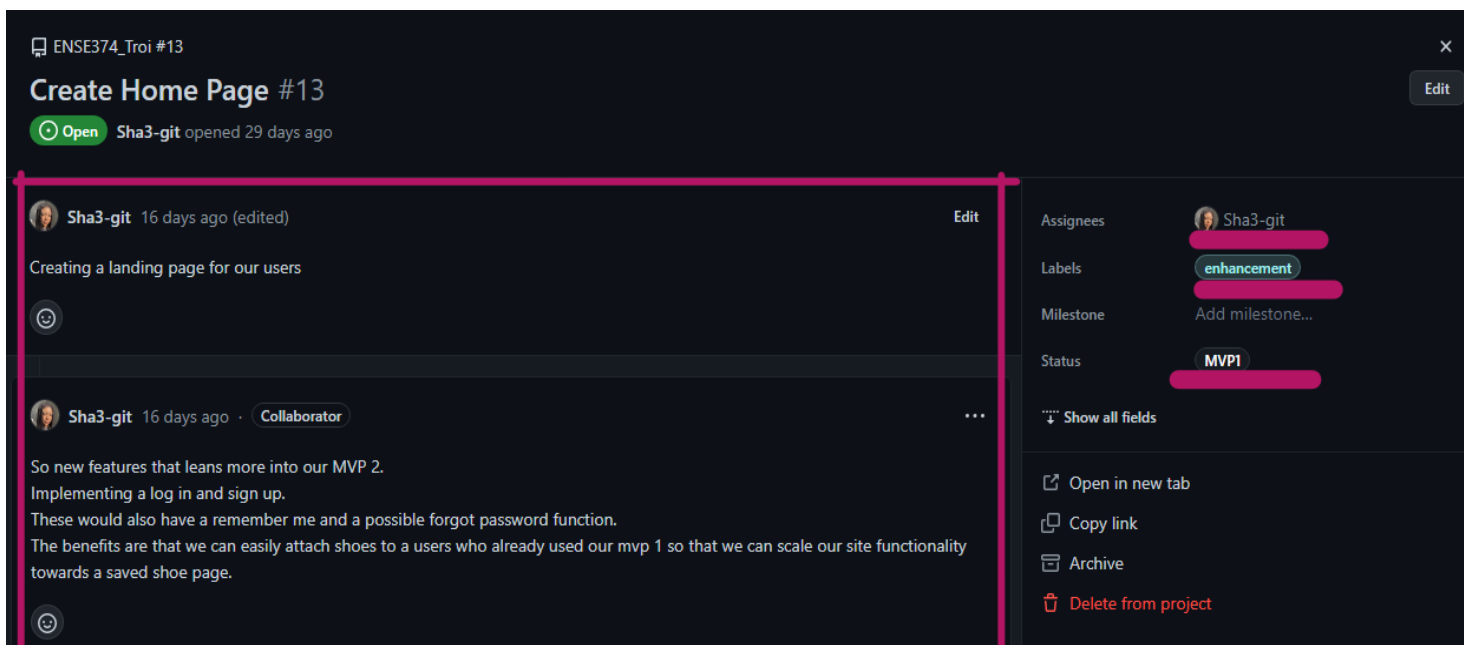
However, implementing a user account gave us a good foothold for our second minimum viable product which would require the use of user accounts for the purposes of saving their preferred shoes and so on.

Our second minimum viable product would mostly involve standardizing our UI with cards and a carousel, and implementing key functionalities that would let the user delete their account, recover their password, use a remember me function, and have access to their own profile. Our contact page is currently not functional so our second minimum viable product would also include implementing functionality for it. A company email has been set up and the necessary dependencies have also been researched in order to implement this. Additionally, we would fix hiccups in our responsive pages (three pages have poor responsiveness) by rendering certain pages. For example, the price page will be rendered with a different UI that would work better on a mobile device. We would also optimize our front-end loading time with regards to how we render the page. Currently, our page loads quickly if the user has fast internet speeds, but upon throttling our page's network it becomes painstakingly slow to render at slower speeds.



For example, if the user was using high traffic internet or using their data, they would run into this issue. There is also the issue where the page has buffering/flashes whenever requests are made, despite fast internet speeds. Therefore, our second minimum viable product will firstly render all the shoe options pages into a singular page through the use of a carousel which would allow us to hierarchically render each option page based on priority; this way, the user can interact with our site while assets are still loading from the back end. Secondly, user's decisions will be saved locally in their browser and then sent after they have chosen all of their shoe options. These changes will optimize our site's controller and view.

#### IV) Project Development

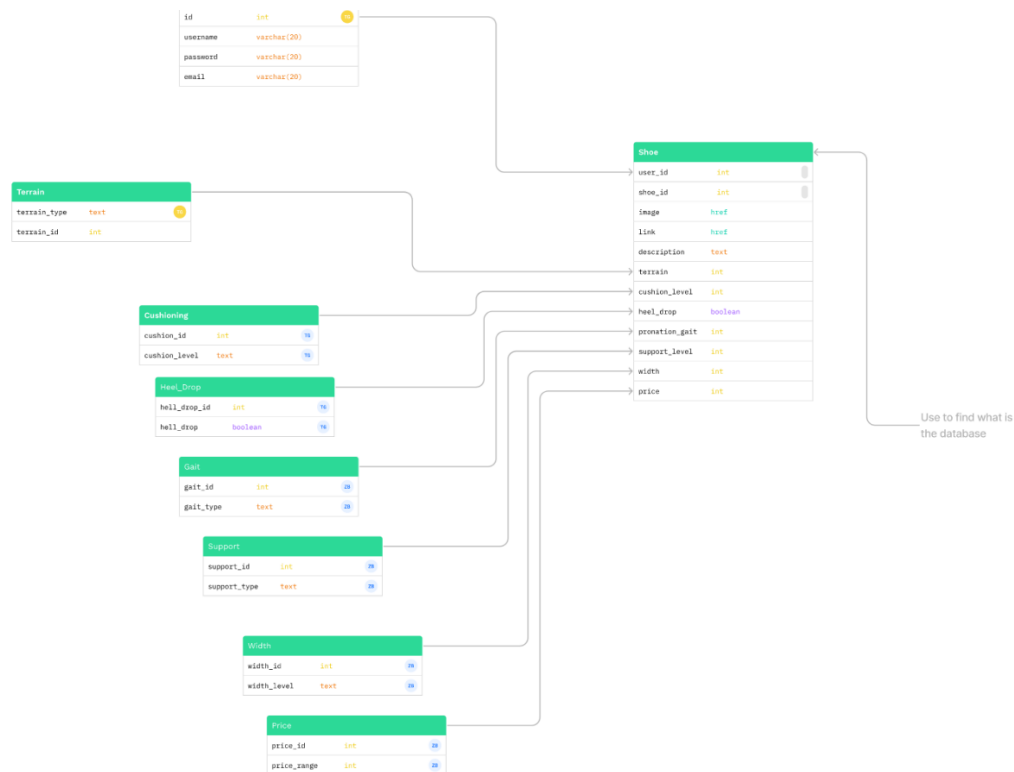


**Figure 5:** *Kanban utilization diagram*

Our Kanban was a key aspect in our project development as a forum for discussion and error diagnosis. We declared our minimum viable product components as issues and created an efficient process to discuss development components, document milestones, seek help during hurdles, and create a centralized zone for information. Irrelevant issues were closed or replaced with improved issues that reflected our MVP envisioning. Furthermore, tags such as “to-do” indicated the active tasks, and the “enhancement” tag indicated improvement opportunities for future MVPs.

Since the backend was the framework used to structure the controller-model interaction, we placed emphasis on the design and discussion of UML diagrams. Therefore, programming

the back-end, which is often a challenging component in development, went seamlessly. The integration was also seamless because there was a clear understanding of our logic before implementing the programming. To abide by the “Three Normal Form” standard for databases, we separated repeated data into separate collections; this eliminated redundancy and difficulty of upscaling our project.



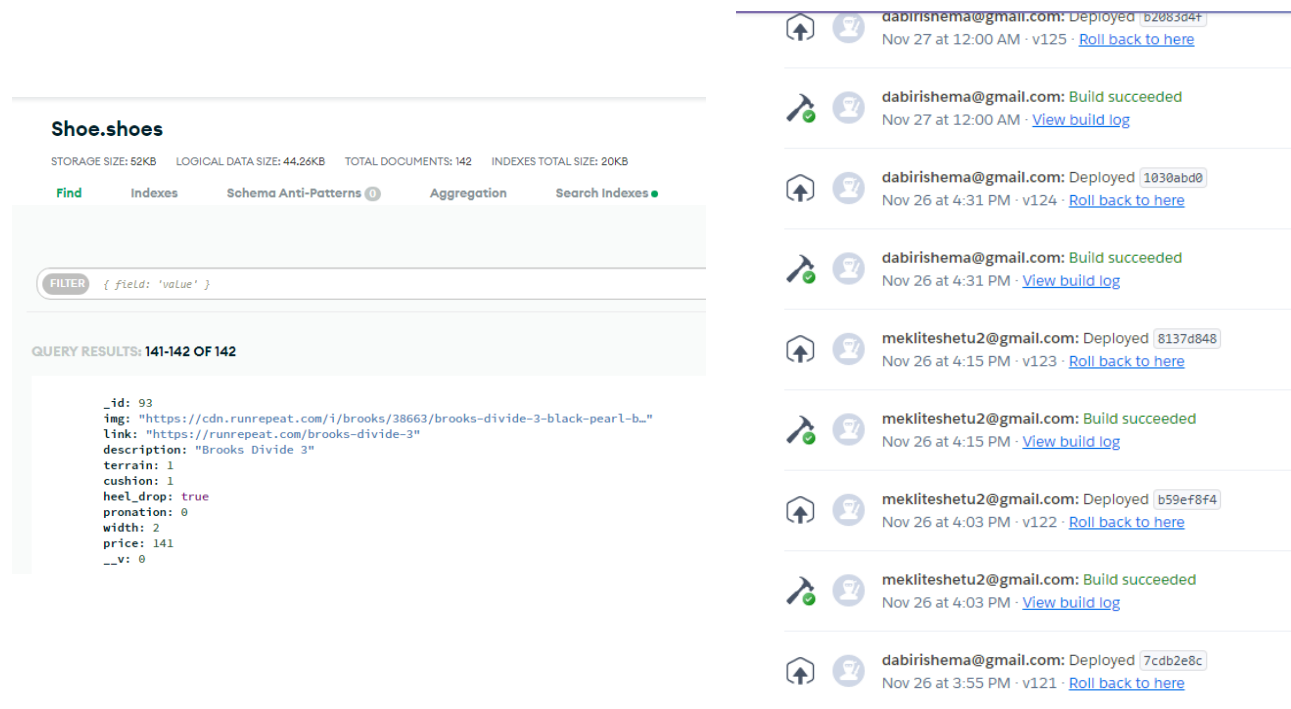
**Figure 6: MVP 2 UML data diagram**

If an addition or removal of a specific shoe parameter was required, each entry in the database needs to be updated. However, with our setup, we only change one row in a singular sub-collection that all the shoes can be referenced to. For example, if we need to replace gravel terrain with synthetic track, we need only change the third entry in the terrain collection. This localization of database dependencies reduces such redundancy. The Model-View-Controller architecture provides a high-level overview for software development, but there were limitations when describing database organization and implementation. These diagrams provided visualization for the proposed implementation and benefits. Our MVC diagram was crucial in expanding our understanding of code component interactions; it influenced our process of task prioritization and separation of concerns. (discuss user questionnaire)

The Node.js application was deployed using Heroku server. Heroku provided a conducive source control feature that enabled version control and deployment rollback system.



This feature was useful when we ran into merge conflicts we could not resolve; the only solution was deployment rollback. We later became familiar with its branching and check-out systems that provided better experimentation such as looking at desktop and laptop views of our web page. Heroku was chosen over lightsail due to its specificity for dynamic web hosting and understandable interface.



**Figure 7: MongoDB shoes collection entry example      Figure 7: Heroku Deployment example**

We hosted our database remotely on MongoDB, and this process was necessary for hosting our site remotely on Heroku. MongoDB is also convenient as future interested parties would not need to clone over 100 shoes into their local database. It also provides greater convenience in managing our backend with multiple functions. For example, the search index opens doors for more use and functionality of the shoes database if we expand our project scope for future MPVs. The database framework is established to enable easy expansion or reduction of project scope. Extra brands, demographics, and shoe types can be integrated without breaking the application functionality. Overall, the hosting decision was pivotal in maintaining separation of concerns among our view, controller, and model.

## V) Programming Principles

Our team displayed great data literacy by implementing the best combination of data sources with their corresponding data type. Efficient use of variable scope is displayed by

encapsulating variables within their relevant methods. Most variables' lifespan never exceed the scope of their use. However, our naming conventions and documentation could be clearer. For example, there is a combination of various single-letter variable names used as temporary values and variables implemented with proper conventions. This naming mixture might viewers during their initial assessment of the code.. We could have also utilized more descriptive documentation in our front end as there are numerous classes and divs whose purpose are not well defined and their removal would break our view. These are a few components to improve for future implementation.

## **VI) Conclusions and Reflections**

As a team, we felt proud and ecstatic with the project accomplishments. We faced several hurdles including merge conflict, responsive design failure, plan changes, dysfunctional features, and limited time, but we were able to overcome all roadblocks. To resolve these problems, we expanded our knowledge beyond the scope of class topics, and these skills will greatly benefit us in industry. Our team was successful in meeting our goals as we set them. The communication between active team members was clear and concise in a scrum-like manner; our weekly check-ins were very beneficial in keeping us on track. There was nothing to dislike in terms of our project as it was something that, by the end of completion, we had all grown to be passionate about. As a team, we are most proud of the organization and commitment of our participating members to produce a high-quality deliverable. Our product and documentation reflects our hard work and determination for this project.

Individually, we learned that perfect was the enemy of good; it is better to lay initial groundwork and develop enhancements later. We connected with experts and establishments in the community such as the Running Room and Foster's Shoes. We experienced first-hand the technology and systems currently used to solve problems similar to what Curre aims to resolve.

We will be using the domain-driven design as a high level approach to developing applications as well as open source libraries such as bootstrap. In the future, we are looking into developing or utilizing more libraries. What we would have liked more help with were more in-class tutorials about source control and dealing with merge conflicts for larger projects. Information about dynamic web hosting on cloud hosting services would have also been highly beneficial. This way, we could get up-to-date information about these services and apply them when working in the industry.