**Show Me Shoe**

Asim Ghimire, Bishrav Shiwakoti, Mallika Balayar, Shubhi Manandhar

Bsc. (Hons.) Computing, Softwarica College of IT and E-commerce, Coventry University

ST4008CEM Computing Activity Led Learning Project 1

Giriraj Rawat

February 28, 2025

Table of Contents

[Table of figures 3](#_Toc191382926)

[Introduction 4](#_Toc191382927)

[Figure 1: 4](#_Toc191382928)

[Aims 5](#_Toc191382929)

[Objectives 5](#_Toc191382930)

[Problem statement 6](#_Toc191382931)

[Features 7](#_Toc191382932)

[Software Requirement Specification 8](#_Toc191382933)

[Functional Requirements 8](#_Toc191382934)

[Non-Functional Requirements 9](#_Toc191382935)

[ Usability 9](#_Toc191382936)

[ Security 9](#_Toc191382937)

[ Reliability 9](#_Toc191382938)

[ Performance 9](#_Toc191382939)

[ Availability 10](#_Toc191382940)

[ Scalability 10](#_Toc191382941)

[Figure 3: 10](#_Toc191382942)

[Scope 12](#_Toc191382943)

[Development Methodology 13](#_Toc191382944)

[Methodology 13](#_Toc191382945)

[Tools and Technologies 14](#_Toc191382946)

[Conceptual Diagrams 16](#_Toc191382947)

[*Entity Relationship Diagram* 16](#_Toc191382948)

[*Use Case Diagram* 17](#_Toc191382949)

[System Architecture 18](#_Toc191382950)

[Project Plan 19](#_Toc191382951)

[Prototypes 20](#_Toc191382952)

# Table of figures

[Figure 1: 4](#_Toc191387790)

[Figure 2: 6](#_Toc191387791)

[Figure 3: 10](#_Toc191387792)

[Figure 4: 11](#_Toc191387793)

[Figure 5: 12](#_Toc191387794)

[Figure 6: 13](#_Toc191387795)

[Figure 7: 15](#_Toc191387796)

[Figure 8: 17](#_Toc191387797)

[Figure 9: 18](#_Toc191387798)

[Figure 10: 19](#_Toc191387799)

[Figure 11: 20](#_Toc191387800)

[Figure 12: 21](#_Toc191387801)

[Figure 13: 21](#_Toc191387802)

[Figure 14: 22](#_Toc191387803)

[Figure 15: 22](#_Toc191387804)

[Figure 16: 23](#_Toc191387805)

[Figure 17: 23](#_Toc191387806)

[Figure 18: 24](#_Toc191387807)

[Figure 19: 24](#_Toc191387808)

[Figure 20: 25](#_Toc191387809)

[Figure 21: 25](#_Toc191387810)

[Figure 22: 26](#_Toc191387811)

[Figure 23: 26](#_Toc191387812)

[Figure 24: 27](#_Toc191387813)

[Figure 25: 27](#_Toc191387814)

[Figure 26: 28](#_Toc191387815)

[Figure 27: 28](#_Toc191387816)

# Introduction

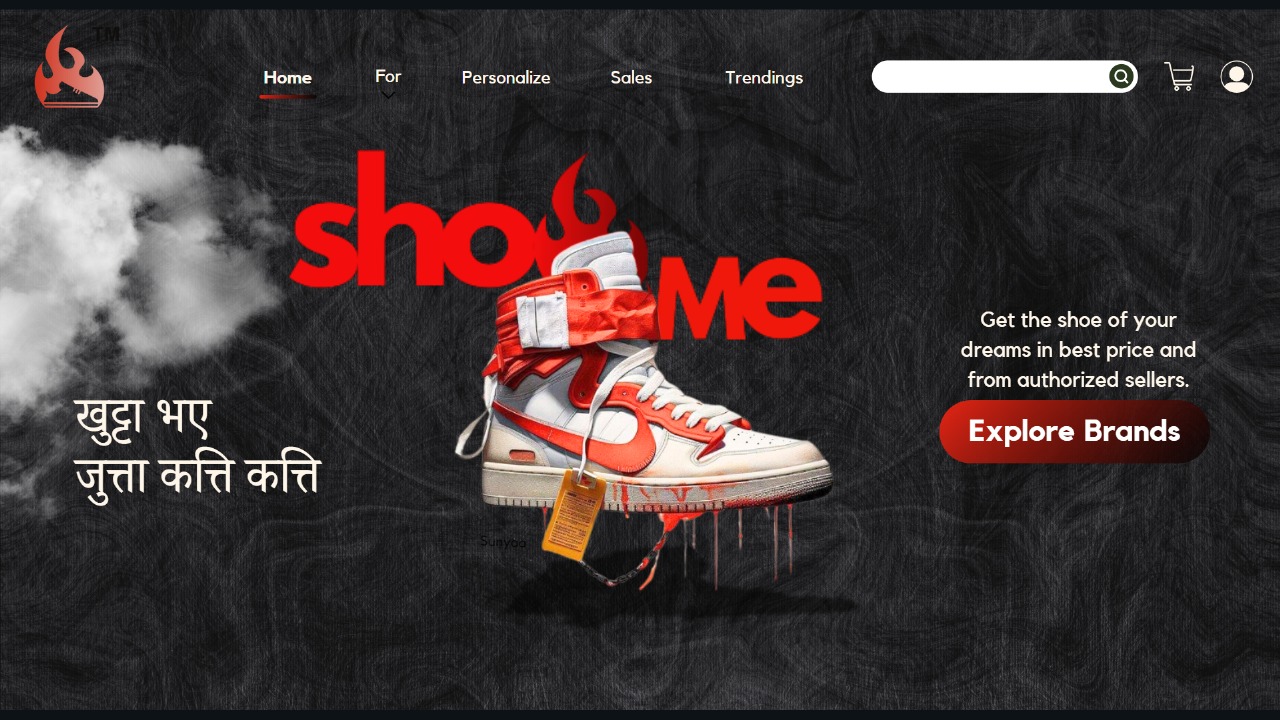
Show Me Shoes is an e-commerce platform which stands out in the huge corrupt market, where high copy shoes are sold to innocent buyers at the rate of original product. This will be possible as Show Me Shoes will collaborate with real brands to handover the product to the consumers without involving any kind of mediators.

Along with that, Shoe Me Shoes will also make the purchasing experience of shoe enthusiasts easier and more efficient with its user-friendly nature. Customers can simply put the shoe to the cart then place an order while tracking their order in real time. The website will also show what shoes are trending at the moment, have different sections for men, women and kids, will let the customers customize the shoe as their liking and consist of many other interesting features.

Moreover, for the managing data, allowing for the creation, reading, updating, and deletion of records in the database, the system will include CRUD. These data will be helping the system to run different operations while looking after the flawless functioning of the website.

## Figure 1:

*Dashboard for ShowMeShoe*



# Aims

Show Me Shoe is software that is aimed to overcome the hazels of finding true authorized sellers of branded shoes in Nepal with providing a user-friendly platform to ease the process of buying them. With the growth of the website, Shoe Me Shoes also focuses on expanding its domain to other nations.

# Objectives

* Make the online shopping process easier with user-friendly interface.
* Promote international as well as national brands.
* Reduce operational costs compared to traditional retail methods.
* Add real-time updates without hindering previous functions.
* Safe, Strong and Simple transactions.
* Providing secure authentication and authorization to its users.
* Gathering customer data to make better marketing strategies and getting more users.

# Problem statement

In Nepal, there are only few authorized shops of branded shoes and people living far cannot physically access to them when they want to. Moreover, there are high chances the retailers other than that of authorized shops with scam customers by selling high copy shoes at the price of real branded shoes.

To address these issues, a well-equipped and trustworthy shopping platform is required, providing it users with convenient, easily accessible and versatile features. By incorporating technologies like Python, Django and SQLite3, an extensive e-commerce website can be built to enhance user experience by lessening the time and effort in finding right shoe, and ensure the payment methods are trustworthy and safe.

Figure 2:

*Shoe Me Shoe (Problems and Solutions****)***



# Features

* Order shoes remotely from the website.
* User account management.
* GSAP is providing wide range of animation. (for example: in our log in page, the shoes are hovering around in the screen which is done by using GSAP)
* Locomotive scroll is giving parallax effect. (Smooth scrolling of pages in the website.
* Secure Transactions.
* Drop down feature.
* Interactive Designs.
* Search bar for finding the shoes.
* Different Sections for Men, Women, and Kid shoes along with trendy once.

# Software Requirement Specification

A **Software Requirements Specification (SRS)** is a detailed document that defines both the functional and non-functional requirements of a software system. It outlines the system's expected features, behaviors, and limitations, ensuring all stakeholders, including clients, developers, and project managers, are aligned on what the software should accomplish.

## Functional Requirements

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So it’s essential to make them clear both for the development team and the stakeholders (AltexSoft, 2023).

* User access: Users can sign up with new account and login if they already own an account in the website.
* Shoes Information (Example: available sizes and colors)
* Real-time availability of shoes in website: Users will be informed if the shoe is sold out or not.
* Order generation.
* Order cancellation.
* Order returning.
* Notification system: when order is conformed, cancelled, delivered or delayed due to some reasons, the customer will be notified via email which they logged in through.
* Generate updates (for example: Adding new shoes timely to men, women, kids and trending sections.)

## Non-Functional Requirements

Nonfunctional requirements are not related to the system's functionality but rather define how the system should perform. They are crucial for ensuring the system's usability, reliability, and efficiency, often influencing the overall user experience.

### Usability

Usability defines how difficult it will be for a user to learn and operate the system. We can assess usability from different points of view:

**Efficiency of use**: the average time it takes to accomplish a user’s goals, how many tasks a user can complete without help, the number of transactions completed without errors, etc.  
  
**Intuitiveness**: how simple it is to understand the interface, buttons, headings, etc.  
  
**Low perceived workload**: how many attempts users need to accomplish a particular task.

### Security

Security requirements ensure there is protection from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different user roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information. Security also includes protection against viruses and malware attacks.

### Reliability

Reliability defines how likely it is for the software to work without failure for a given time. Reliability decreases because of bugs in the code, hardware failures, or problems with other system components.

### Performance

Performance is a quality attribute that describes the system's responsiveness to various user interactions. Poor performance leads to a negative user experience. It also jeopardizes system safety when it’s overloaded.

### Availability

Availability reflects the time that the system’s functionality and services are available for use with all operations. So scheduled maintenance periods directly influence this parameter. And it’s important to define how the impact of maintenance can be minimized. When writing the availability requirements, the team has to define the system's most critical components that must be available at all times.

### Scalability

Scalability requirements describe how the system must grow without negatively impacting its performance. It means serving more users, processing more data, and doing more transactions. Scalability has both hardware and software implications. For instance, you can increase scalability by adding memory, servers, or disk space.

## Figure 3:

*Non-Functional Requirement*

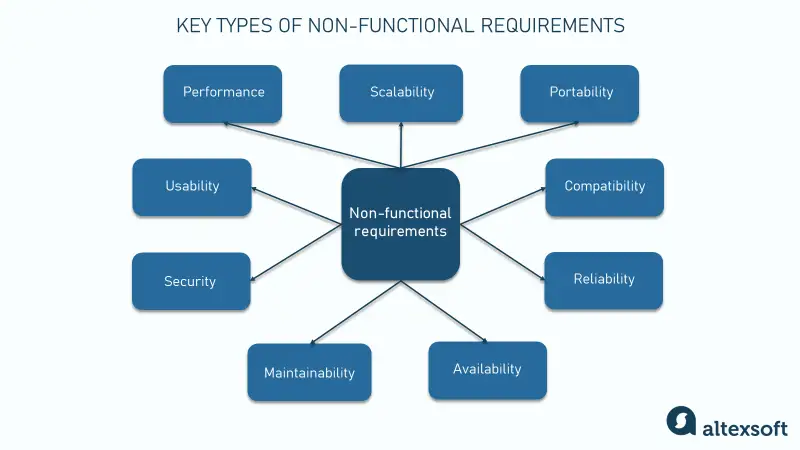
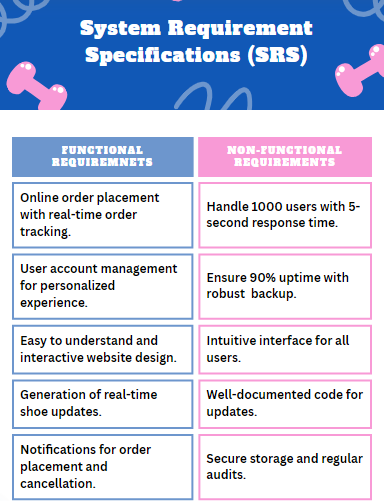


Figure 4:

*System Requirement Specifications*



# Scope

• Order placement: Customers can place order from various brands.

• Easy Shoe Finding: Customers can find shoes via individual sections like men, women, kids and trending or simply search for the shoes from search bar.

• User Management.

• Easy order updates.

• Order Cancellations.

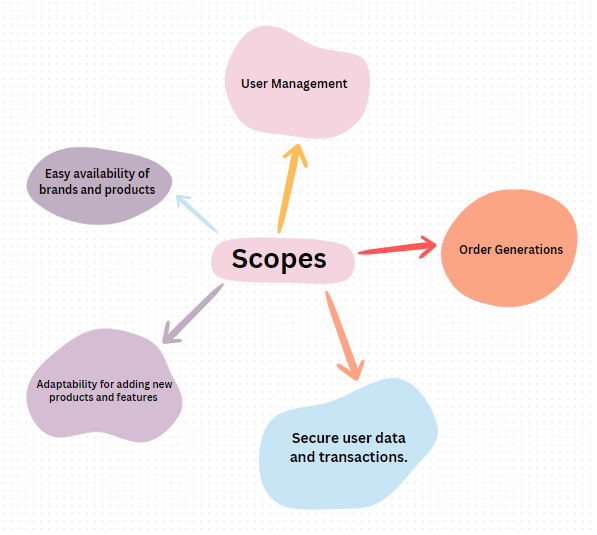
• Integration with SQLite3.

• Scalability: Designed for future growth with added brands, products and features.

• User Data Security.

Figure 5:

*Scope of the System*



# Development Methodology

Development methodology refers to a set of structured processes and practices that are defined by common characteristics to help streamline and categorize the development process. Any given development methodology sets rules and conditions for communication and the passage of information between team members.

Such methodologies are considered frameworks to organize the working process and plan the next steps in the most appropriate and efficient manner for a project. (LimeUp)

## Methodology

Agile methodology is a project management framework that breaks projects down into several dynamic phases, commonly known as sprints.

The Agile framework is an iterative methodology. After every sprint, teams reflect and look back to see if there was anything that could be improved so they can adjust their strategy for the next sprint.

Figure 6:

*Agile Methodology*



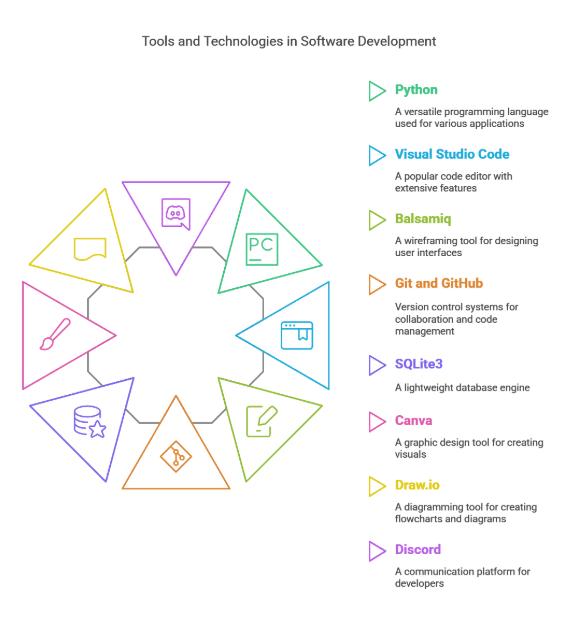
## Tools and Technologies

Software development tools are specialized applications designed to assist developers in building and testing software. These tools are essential for experienced programmers, helping to simplify complex coding tasks and aiding in debugging. Below are some of the tools and technologies commonly used:

* Python
* Visual Studio Code
* Balsamiq
* Git and GitHub
* SQLite3
* Canva
* Draw.io
* Discord
* WhatsApp
* Django
* HTML
* CSS
* Instagram
* Pinterest
* ThreeJS
* Locomotive Scroll

Figure 7:

*Tools and Technologies*



## Conceptual Diagrams

Conceptual diagrams illustrate proposed connections between sources, stressors, and biological reactions.

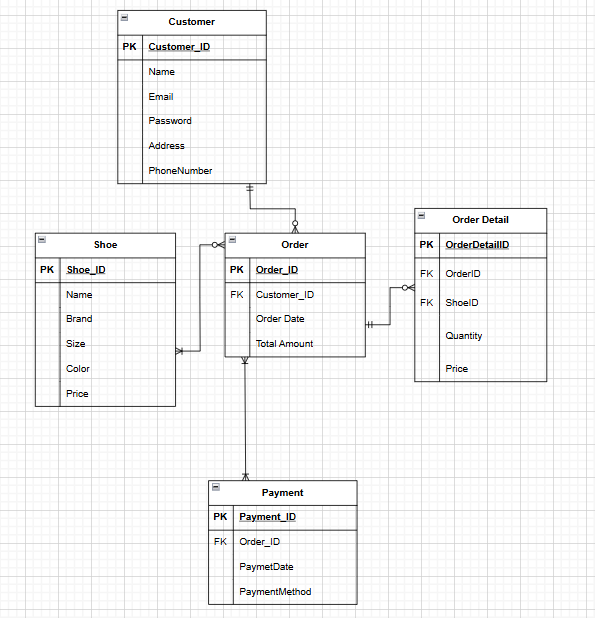
A conceptual diagram serves as a visual depiction of how a system operates. These diagrams are utilized to illustrate proposed connections among sources, stressors, and biological responses in aquatic ecosystems. Conceptual diagrams, along with their descriptive narratives, are valuable tools throughout the Stressor Identification process, aiding in everything from initial brainstorming to establishing a framework for data gathering and analysis, and finally in organizing and presenting findings.

## *Entity Relationship Diagram*

An Entity Relationship Diagram, commonly referred to as an ERD, ER Diagram, or ER model, is a structural diagram utilized in the design of databases. An ERD features various symbols and connectors that illustrate two key pieces of information: the primary entities within the boundaries of the system and the relationships that exist between these entities.

Figure 8:

Entity Relationship Diagram for ShowMeShoe



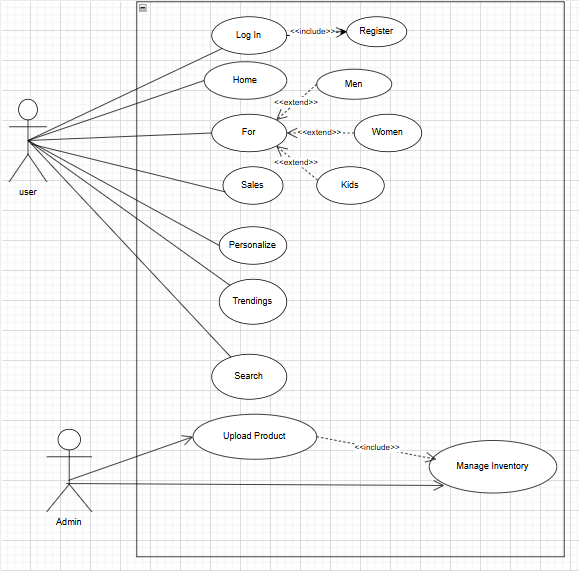
## 

## *Use Case Diagram*

A UML use case diagram serves as the main representation of system or software requirements for a software application that is currently being developed. Use cases describe the anticipated actions (what) without detailing the actual implementation methods (how).

Figure 9:

*Use Case Diagram for Show Me Shoe*



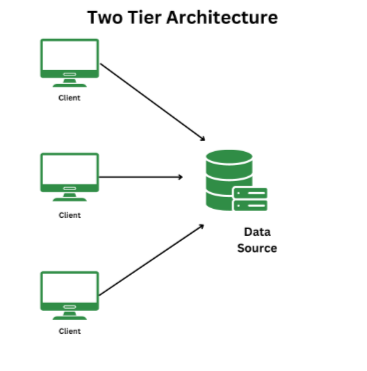
# System Architecture

Software architecture comprises a collection of guidelines that shape the design and development of software. It outlines the organization of the software system and its structural components. Additionally, it illustrates the interactions among different elements, levels of abstraction, and various aspects of the software system.

Architecture can serve to establish the objectives of a project or can assist in steering the design and creation of a new system. Software architecture consists of a framework of principles that govern how software is created and constructed.

Figure 10:

*System Architecture*

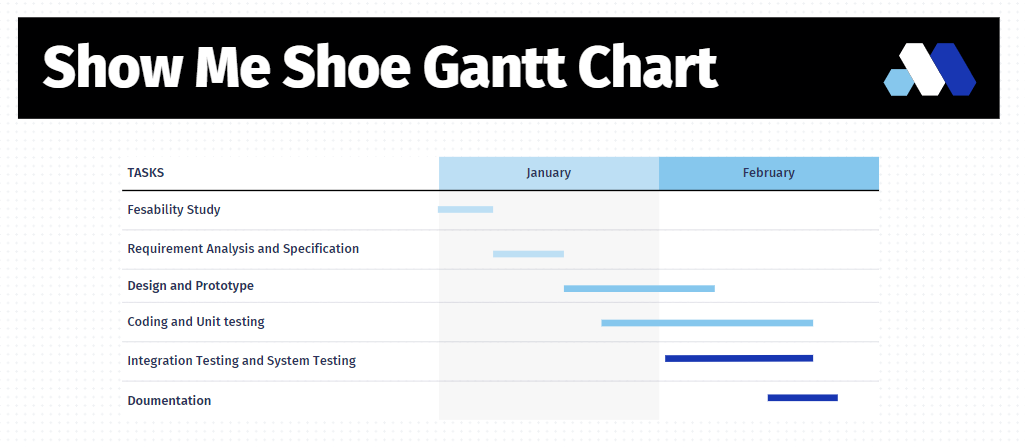


# Project Plan

A project's scope, goals, tasks, deliverables, milestones, communication channels, budget, and deadlines are all spelled out in detail in a project plan, sometimes referred to as a work plan. Providing a clear roadmap or blueprint for project execution, providing early notice of project requirements, and ensuring the project continues on track toward completion are the goals of a project plan, and in fact, of project planning.

Figure 11:

*Show Me Shoe Gantt Diagram*



# Prototypes

An early model, sample, or product release made to test an idea or procedure is called a prototype. In order to increase the accuracy of analysts and system users, a prototype is usually utilized to test a new design. It is the stage in between formalizing and assessing an idea.

All design disciplines use prototypes, which are an essential component of the design process. Before investing in mass production, architects, engineers, industrial designers, and even service designers create prototypes to test their ideas.

The following are the prototypes of Show Me Shoe website with both low-fidelity and high fidelity designs:

Figure 12:

*Prototype - Login Page of User Authentication (high-fidelity)*



Figure 13:

*Prototype - Login Page of User Authentication (low-fidelity)*

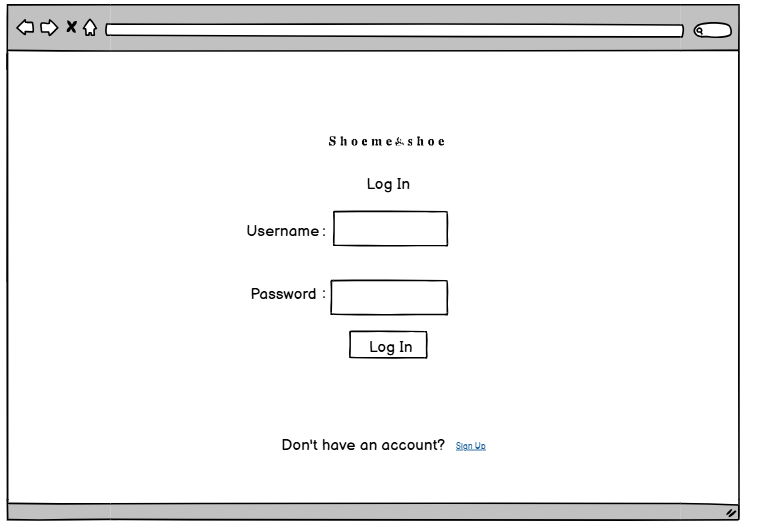


Figure 14:

*Prototype - Signup Page of User Authentication (high-fidelity)*



Figure 15:

*Prototype - Signup Page of User Authentication (low-fidelity)*

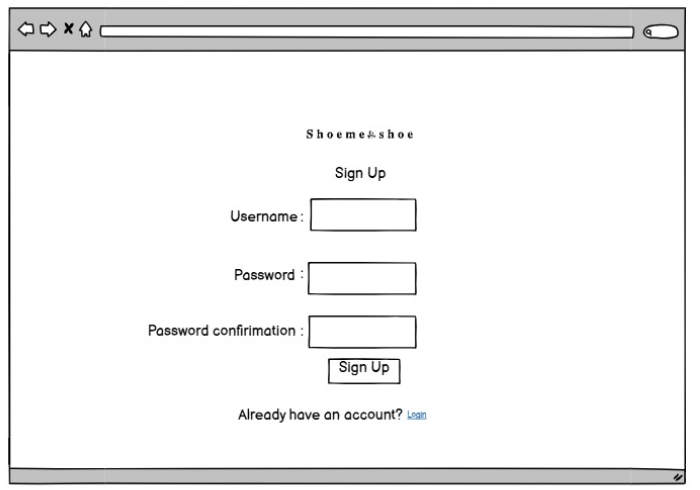


Figure 16:

*Prototype-Username already existing error in signup page (high-fidelity)*



Figure 17:

*Prototype-Username already existing error in signup page (low-fidelity)*

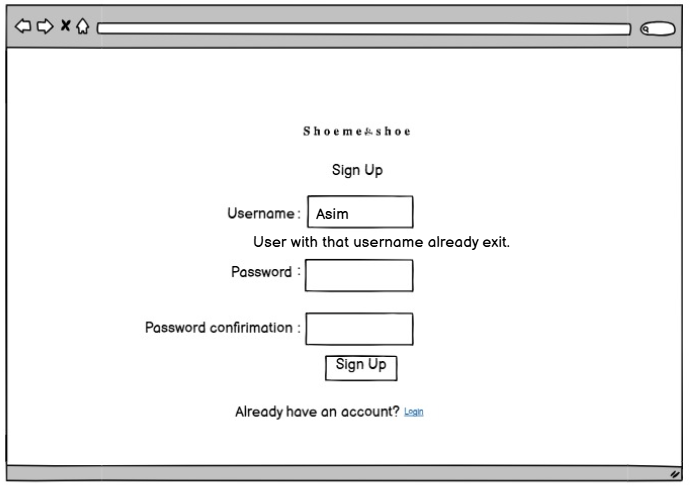


Figure 18:

*Prototype-Required field in signup page (high-fidelity)*



Figure 19:

*Prototype-Required field in signup page (low-fidelity)*

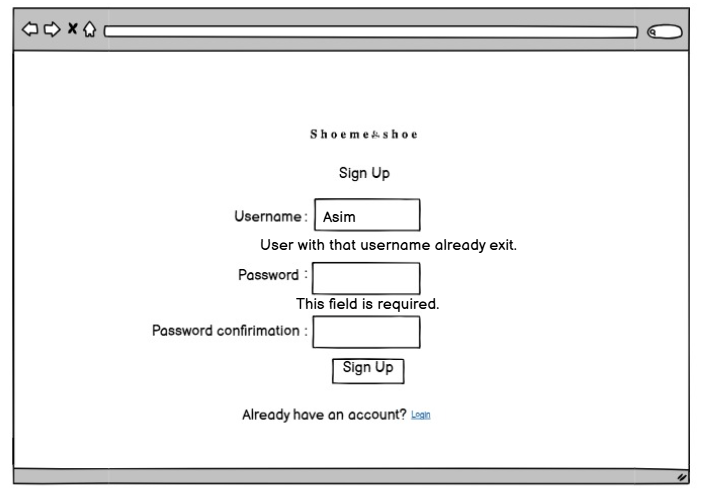


Figure 20:

*Prototype-Dashboard for ShowMeShoe (high-fidelity)*

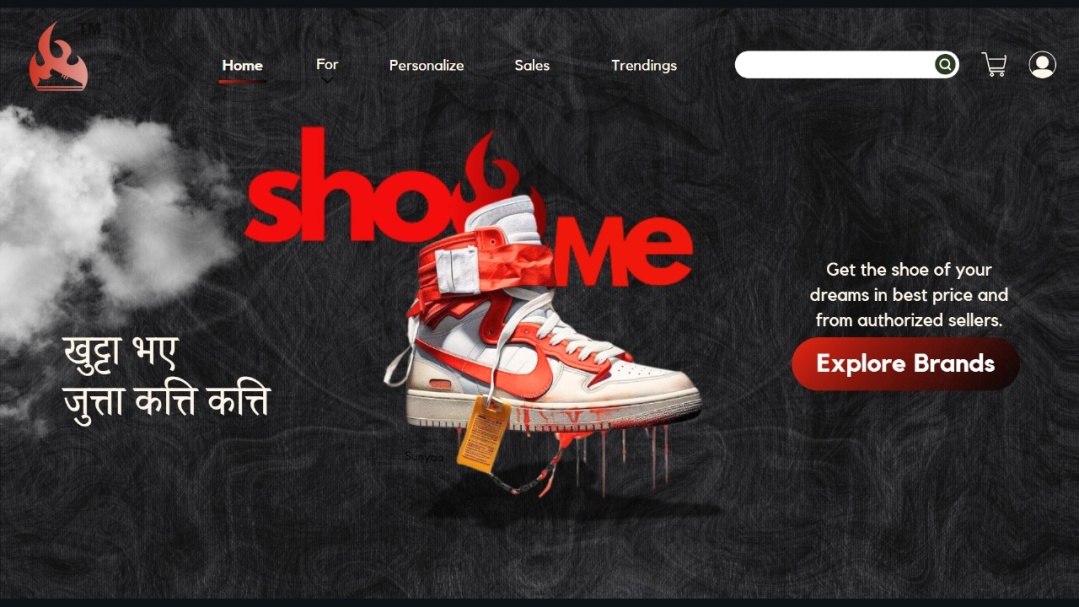


Figure 21:

Prototype-Dashboard for ShowMeShoe (low-fidelity)



Figure 22:

*Prototype-Men section in ShowMeShoe(high-fidelity)*

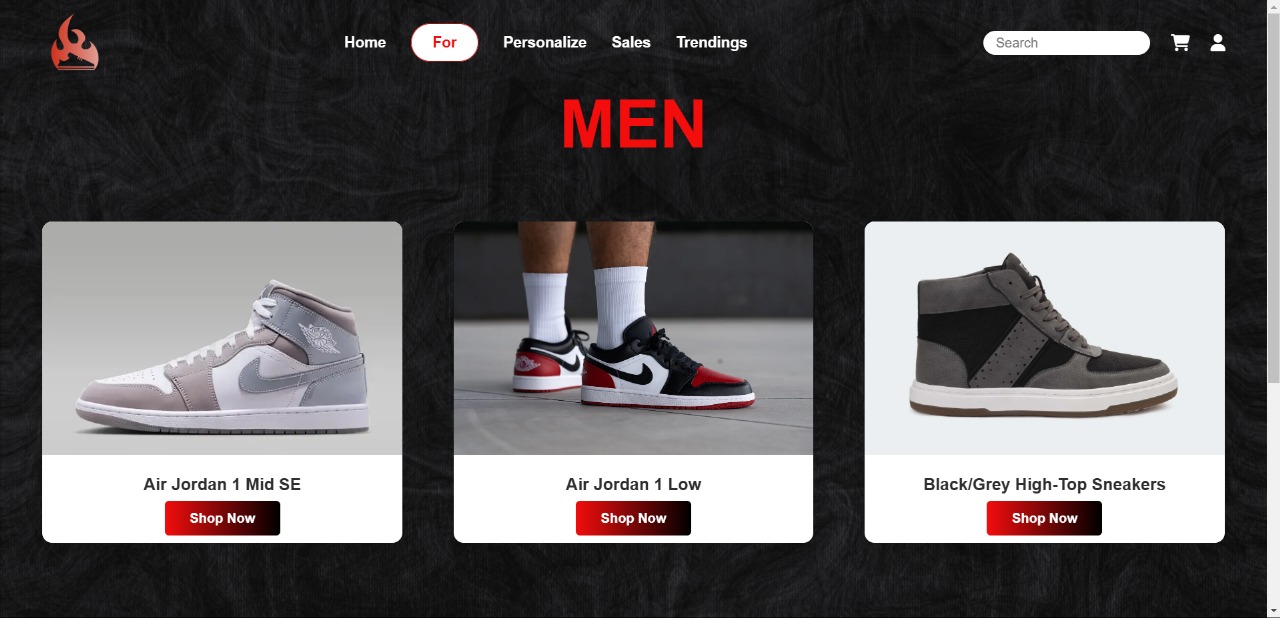


Figure 23:

*Prototype-Men section in ShowMeShoe(low-fidelity)*

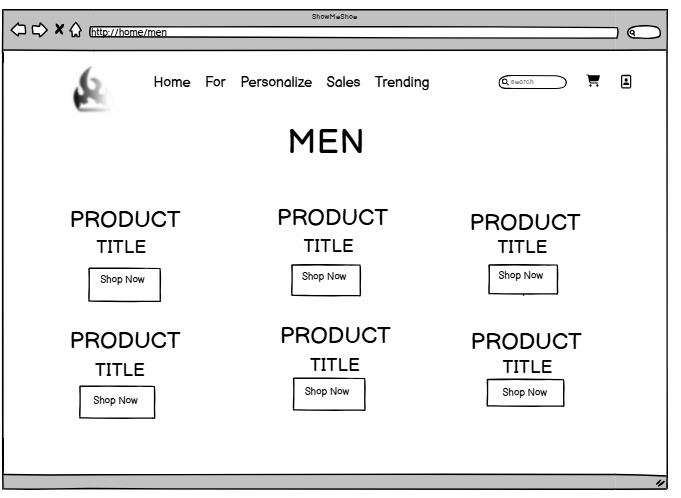


Figure 24:

*Prototype-Women section in ShowMeShoe(high-fidelity)*

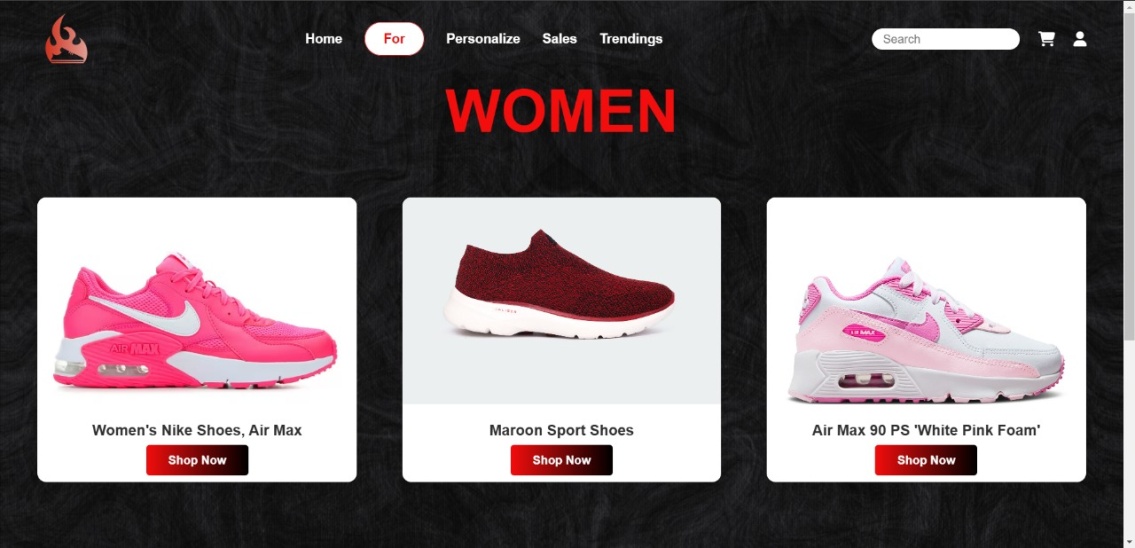


Figure 25:

*Prototype-Women section in ShowMeShoe (low-fidelity)*

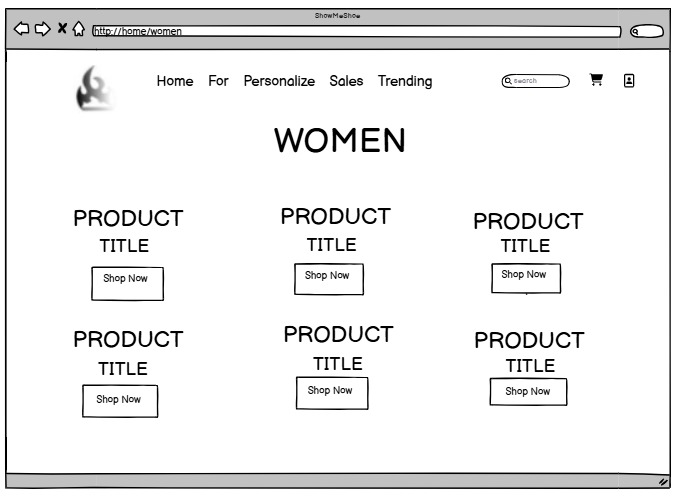


Figure 26:

*Prototype-Kids section in ShowMeShoe (high-fidelity)*

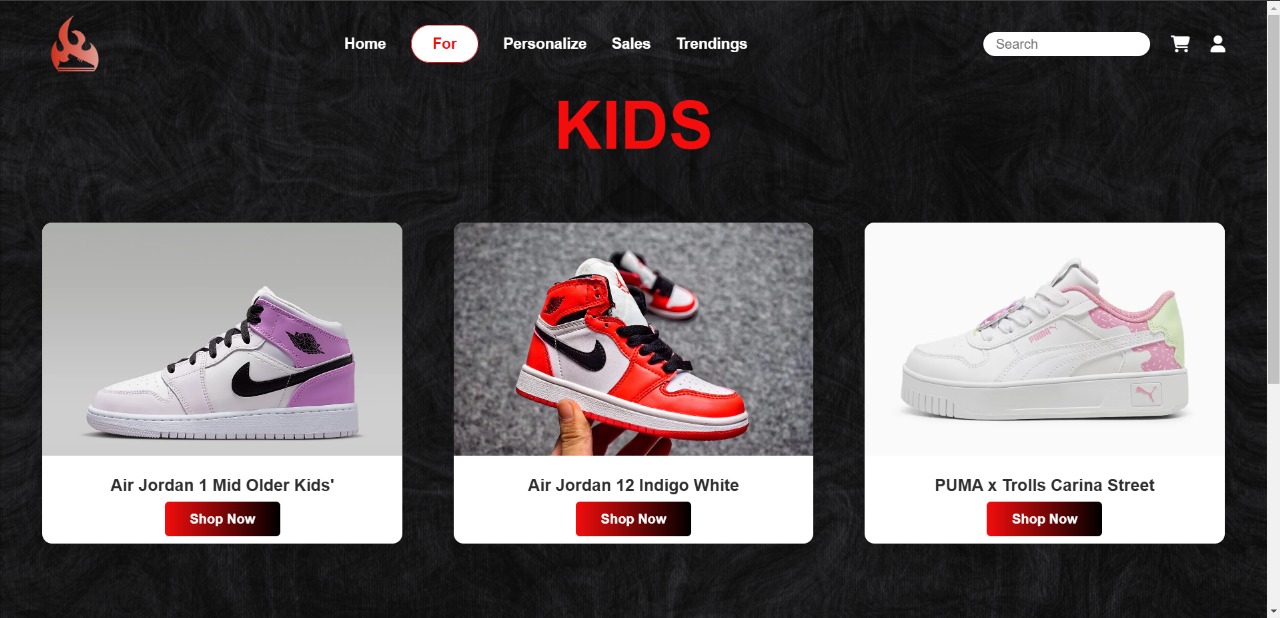


Figure 27:

*Prototype-Kids section in ShowMeShoe (low-fidelity)*

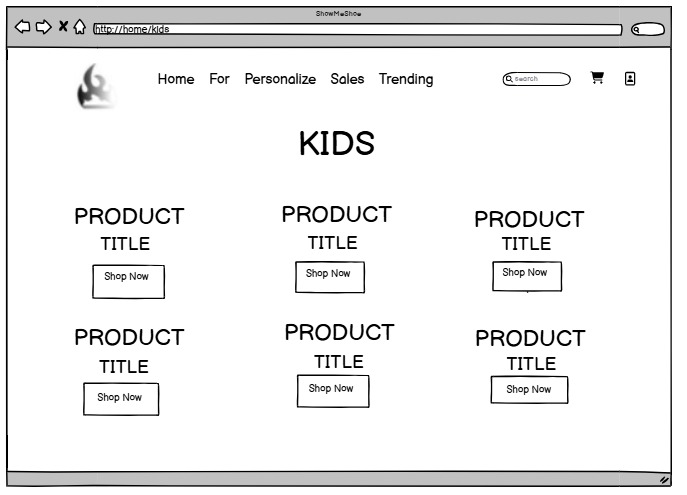


Figure 28:

*Prototype-Trendings section in ShowMeShoe (High-fidelity)*

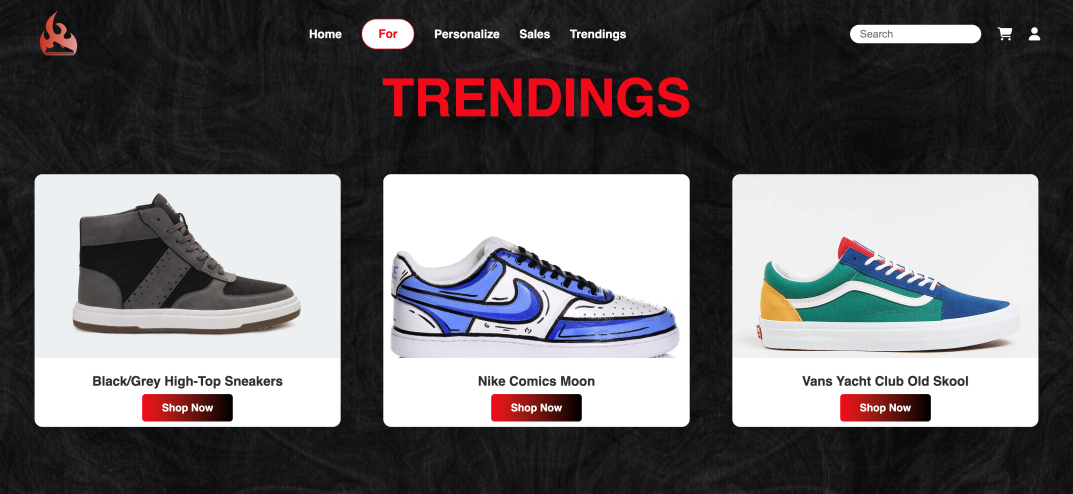


Figure 29:

*Prototype-Trendings section in ShowMeShoe (low-fidelity)*

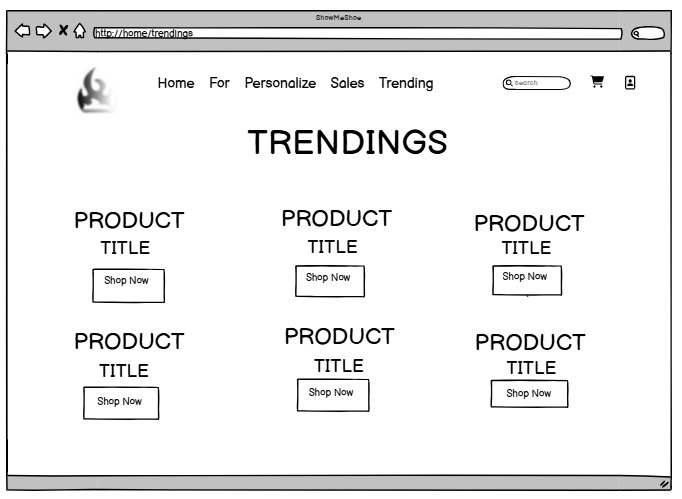


Figure 30:

*Prototype-Sales section in ShowMeShoe(high-fidelity)*

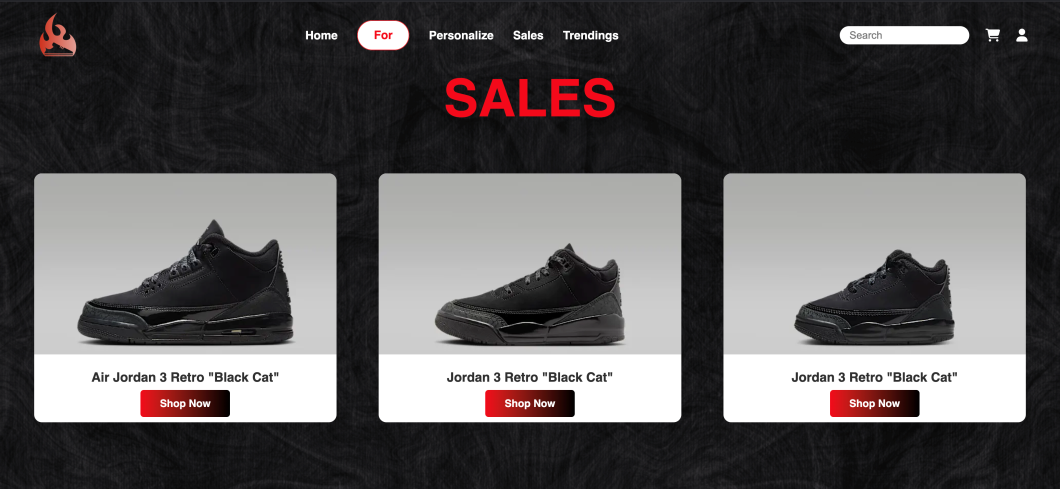


Figure 31:

Prototype-Sales section in ShowMeShoe (low-fidelity)

