Advanced Web Solutions Report: Vendor Application Inventory Platform

SID: 2275875

Introduction

In response to the challenges financial services consultancies encounter in efficiently identifying and selecting technology solutions for clients, Showout has pioneered a revolutionary Vendor Application Inventory Platform. This web application addresses the limitations of the current methodology, which involves multiple document libraries and Excel spreadsheets, resulting in lost revenue, missed opportunities, and inefficient use of time. The platform allows vendors to showcase their services, creating a dynamic marketplace where users can effortlessly find, evaluate, and engage with a diverse array of offerings.

The successful integration of diverse technology solutions in the financial services industry heavily relies on vendor expertise. Finding and selecting the proper companies to deal with might be challenging. It is like dealing with different information sources, collecting data by hand, and not being able to see the whole picture clearly. These obstacles result in knowledge gaps, missed business opportunities, and resource waste. In response to these challenges, the Showout platform is similar to Amazon, specifically tailored for the financial services sector. This platform offers a comprehensive vendor marketplace, advanced search capabilities, detailed vendor profiles, a centralised data repository, and user administration features. Showout's programme streamlines the process of finding and engaging with vendors.

Showout aims to make things work better, help people make smarter decisions, encourage teamwork, increase how easily vendors can be seen, and boost the generation of potential business opportunities. Users possess the ability to effortlessly explore and compare vendors, arrive at informed conclusions, and efficiently join forces. Consequently, vendors acquire enhanced visibility and publicity within the industry.

The development of Showout is a meticulous process that demands attention to security, seamless data integration with existing systems, adherence to industry regulations, and the ability

to scale for future expansion. A comprehensive plan is necessary, including technical specifications, an implementation roadmap, and a cost analysis, to navigate the challenges of enhancing service marketplace visibility, streamlining manual operations, and breaking down information silos that currently lead to revenue losses and inefficiencies. The solution proposed is a Vendor Application Inventory Platform, requiring core features like a centralised data repository, intuitive interface, user profile management, and advanced search functionalities, with the flexibility to add reporting tools, notifications, reviews, and application comparisons. The technology foundation is Django, chosen for its scalability and security, alongside JavaScript, HTML, CSS, and MySQL. User stories guiding the implementation focus on registration, service discovery, vendor management, and comprehensive administrative tools. The platform emphasises essential functionalities while allowing room for optional features and technologies to accommodate diverse needs and future enhancements.

E-commerce today has become a major component of contemporary business landscape. Due to the various alternatives available to consumers and also the quest to access the right information about a product or service, consumers are always in search for platforms that are user-friendly and versatile. The birth of several e-commerce platforms in the business world has been game changer in the e-commerce industry. With pacesetters such as Amazon, Alibaba, etc. they have set a benchmark that entrepreneurs could leverage to make to make a difference in the continuous changing e-commerce market. Given that, this project intends to develop an Amazon-like web application that utilises the unique Django, which has been described as a high-level Python web framework.

Django stands out as a good choice for web application development due to its diverse features designed for rapid development security. Django has been described as a comprehensive framework due to its unique features such as authentication mechanisms, built-in control interfaces which facilitates the development process (Haleem et al., 2022). This Django extension significantly reduces the time and effort required to design and implement complex web applications, making it an ideal solution for our project. Django has been built to follow the Model-View-Controller (MVC) architectural pattern, which improves custom organization and control. This separation of concerns promotes collaboration and further development of work

(Martin-Rios, Erhardt and Manev, 2022). Given the collaborative nature of our project and the need for iterative development based on user feedback, Django's MVC paradigm aligns with our main objective of this project.

The selection of Django as the primary framework for developing our Amazon-like web application is due to its unique features as intimated earlier. Leveraging Django in this project led to the development of a compelling web application that allows vendors to display their products and services whilst facilitating convenient transactions for consumers. Throughout this project, we explored Django's features, best practices for web application development and demonstrated how Django was used in the project to realize our vision effectively.

1.1 Project aim

The purpose of the project is to design and implement a Vendor Application Inventory Platform that addresses the aforementioned challenges faced by financial services consultancies.
The platform offers the User a user-friendly interface for data entry, maintenance, and retrieval with the goal of centralizing and optimizing the administration of vendor and product information.
Below is a list of specific objectives of the project

- Create user interface
- Create a database to store data of vendors and their products
- Establish user permissions and roles
- Provide users with the ability to search using a variety of parameters and dig down into certain results
- Create connection between users and vendors by storing and providing users with links to vendors websites.
- Provide users with the capability to view attached documents for a vendor or product within the system.

1.2 Significance of the Project.

The development and implementation of the Vendor Application Inventory Platform will help financial services consultancies be able to store all their vendors and product information into a single user-friendly platform for easy managing and retrieval. By capturing and storing vendor applications knowledge in a structured manner, critical information will easily be accessible to authorized users. The centralized approach will also enhance efficiency and productivity since it will reduce the manual effort required for gathering, organizing and analyzing data.

1.3 Resources

To finish the project, the following tools and resources were used:

- PYTHON (version 3.12.1)

  Python is a popular programming language and statistics shows that it is the most widely used programming language. It has a lot of libraries and is the number used in data science. According to statistictimes.com python is the top programming language in TIOBE and PYPL index.

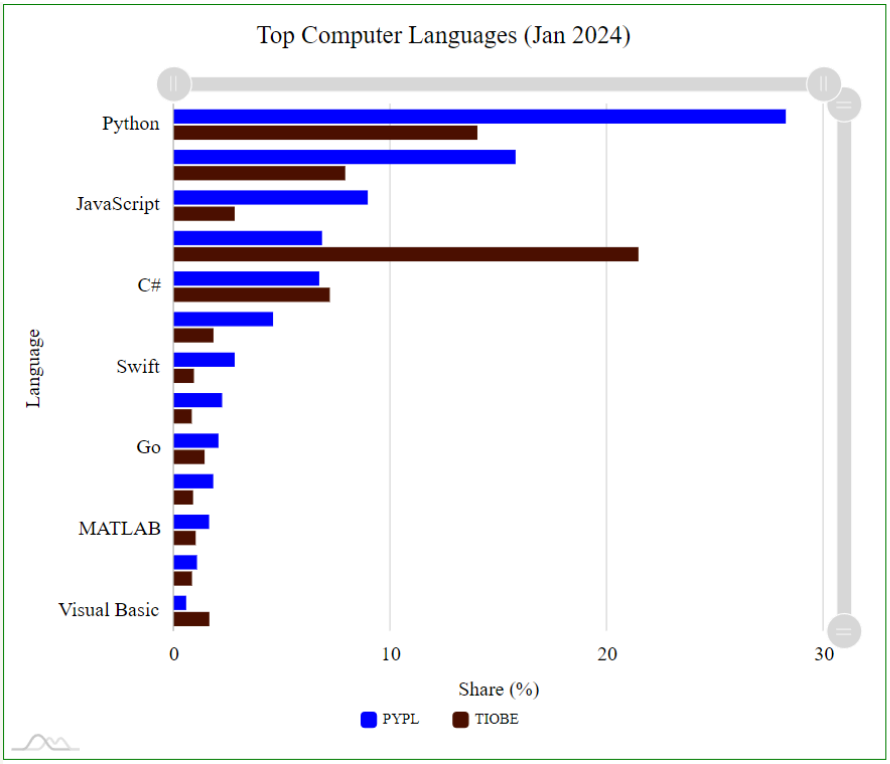


Figure1 shows the top computer languages with python recording the highest.

Source : https://statisticstimes.com/tech/top-computer-languages.php

- DJANGO FRAMEWORK

  Django is an advanced Python web framework that promotes efficient development and simple, straightforward design. It is fast, scalable and secure. [ ].

  Installing django:

- Visual Studio Code
- Android Studio
- GitHub
- Git Bash

- Tech with Tim (Youtube Channel)
- Slack
- CodeWithSteps
- Bootstrap
- DataTables.net
- Adobe Photoshop

1.4 Team Meeting Schedules

The project was managed and completed by all team members. The team communicated on campus during class breaks and using Google Meet. Meetings were held from 6:00 pm every Friday, Saturday and Sunday via Google Meet. The team will watch tutorials on youtube to understand and read other resource material to keep ahead of the lectures and turn to the instructor for coaching and guidance.

1.5 User Stories

1.5.1 Customer user stories

- As a customer I want access to a user-friendly home page where I can explore and compare different technology solutions offered by vendors, helping me to make informed decisions about which solutions best suit my business needs.
- As a customer I want to sign up into the system
- As a customer I want to update and edit my profile
- As a customer I should be able to reset my password
- As a customer i should be able to logout
- As a customer i should be able to delete my account
- As a customer i should be able to add a service to my request
- As a customer i should be able to remove a service from my request list
- As a customer i should be able to view top reviewed services
- As a customer i should be able to view services base on categories
- As a customer i should be able to view all the vendors on the platform
- As a customer i should be able to view attachment on each services
- As a customer i should be able to access vendor website
- As a customer i should see my request history

- As a customer I need the ability to easily search for specific types of technology solutions within the platform, saving me time and effort in the selection process.
- As a customer, I expect the platform to provide detailed information about each technology solution, including service details, customer reviews, vendor contact details etc. allowing me to thoroughly evaluate my options before making informed decisions.
- As a customer I want to rate the vendors' service(s) and be able to also post a comment

1.5.2 Vendor User stories

- As a vendor, I should be able to add service
- As a vendor, I should be able to edit service
- As a vendor, I should be able to delete my service
- As a vendor, I should be able to have a dashboard with some analytics about my services
- As a vendor, I should be able to see list of service
- As a vendor, I should be able to see list of customers
- As a vendor, I should be able to see list of request
- As a vendor, I should be able to export some of my data in a form or pdf, etc
- As a vendor, I need the ability to create an account and setup my profile
- As a vendor, I should be able to login
- As a vendor, I should be able to logout
- As a vendor, I should be able to update my profile details
- As a vendor, I should be able to delete my account

1.5.3 Admin User Stories

- As an administrator, I should be able to login
- As an administrator, I should be able to logout
- As an administrator, I should be able to update my profile details
- As an administrator, I should be able to manage customers
- As an administrator, I should be able to manage vendors
- As an administrator, I should be able to manage country
- As an administrator, I should be able to manage categories

- As an administrator, I should be able to manage service
- As an administrator, I should be able to manage vendor service
- As an administrator, I should be able to manage gender
- As an administrator, I should be able to manage reviews
- As an administrator, I should be able to manage wish list/request
- As an administrator, I should be able to manage groups
- As an administrator, I should be able to manage users

Literature Review

Omotunde and Ahmed's 2023 research article focuses on database security mechanisms, including authentication, access control, encryption, auditing, intrusion detection, and privacy-enhancing strategies. The academic review effectively identifies the domain's current issues and vulnerabilities, as well as viable mitigation solutions. The study looks at new things happening in big data, cloud storage, and connected devices (the Internet of Things). It says the most important thing is to have super strong security, follow clear rules for checking things (auditing standards), use special coding to keep information secret (encryption), and have good ways to make sure only the right people can access things (authentication and access control). Moreover, the insights presented in this research paper serve as a highly valuable resource for enterprises aiming to augment their comprehension of database security.

Pärn et al.'s 2017 study explores the role of Building Information Modeling (BIM) in facilities management (FM). The research highlights its importance in improving building performance, data management, problem mitigation, defect identification, and enhanced processes. Future directions for BIMFM research include data integration, predictive models, automatic as-built/classification, internet of things, energy management, and augmented/virtual reality.

Amazon's business model, digital strategy, employee turnover rate, and innovation ecosystem governance are analysed in a 2018 paper by Zhu and Liu. The study highlights Amazon's use of technology, delivery systems, and customer service to differentiate itself. It also discusses challenges and potential in India and the importance of managing complementors within innovation ecosystems.

The impact of FinTech developments on the financial sector is explored in Jarvis and Han's 2021 literature review, which underlines their ability to introduce innovative financial products, support financial inclusion, simplify operations or reduce costs. It emphasises the role of alternative lending, payment solutions, blockchain technology, and personalised insurance solutions. The review uses institutional theory to understand social changes driven by FinTech innovations and identifies regulatory concerns. Despite risks like cybersecurity threats and

limited oversight, FinTech has the potential to drive positive social and environmental impacts, contributing to sustainability goals in the banking industry. Future research questions cover market equilibrium, regulatory sandboxes, cybersecurity risks, risk-taking, social acceptance, and ageing societies.

A cloud-based inventory management system, designed and implemented by Adegbaju and Odun-Ayo (2020), addresses challenges in maintaining ideal stock levels for Maintenance, Repair, and Operating (MRO) inventory. Traditional systems lack real-time data, leading to increased operational expenses and maintenance efforts. The proposed solution utilises cloud computing, offering stress-free updates, background support, and simplified user interfaces without extra personnel or specialised hardware. The system, developed with Django, deployed on Google App Engine, and using Google Cloud SQL and Storage, aims to optimise Supply Chain Management and reduce operating costs. The study highlights cloud computing's potential to enhance productivity, save costs, and provide flexibility for organisations, particularly those with MRO inventory needs.

Django poses several qualities that make it suitable for developing e-commerce platforms like Amazon. According to Garcia et al. (2017), Django's "battery package" concept provides features for web development such as user authentication, control and form processing. This streamlines the development process so developers can focus on implementing specific features for their e-commerce platform. Django also follows the Model-View-Controller (MVC) design pattern, which improves code organization and control. Forte (2016) shows how Django's MVC model encourages idea sharing among developers by separating concerns about data structure, user communication and business logic. This flexibility allows for easy upgrade and expansion as your e-commerce business grows.

The Django framework has been touted for its enhanced security feature. Tadhani et al. (2024) reported on Django's protection against vulnerabilities such as cross-site scripting (XSS) and SQL injection protection. Django's authentication system also provides mechanisms to securely manage user accounts and permissions and protect sensitive information such as financial transaction and personal information.

While Django may have its advantages, developing an e-commerce platform using Django comes with certain challenges and assumptions. One of the challenges is being able to manage the different aspects of traffic and activity. Verma (2023) discusses strategies for enhancing Django applications for growth such as using caching, using asynchronous work queues and leveraging scaling strategies. Another consideration is the integration of third-party services and APIs essential to e-commerce, such as payment gateways, delivery centres and recommended products. Kelton (2021) emphasize the importance of choosing reliable third-parties and carefully managing dependencies to maintain compatibility and security of Django projects.

Again, performance management are essential to providing a good user experience in e-commerce. Rabl et al. (2012) recommended the usage of performance profiling tools and optimising queries in databast to identify and resolve problems that may affect application performance. According to Hoan et al. (2023), the built-in admin interface for Django has proven to be very effective in managing site contents, listing of products and the processing of orders. Again, the Django framework also incorporates comprehensive testing strategies that enhances the reliability and functionality of their web applications compared to other frameworks like PHP (Salas-Zárate, 2015).

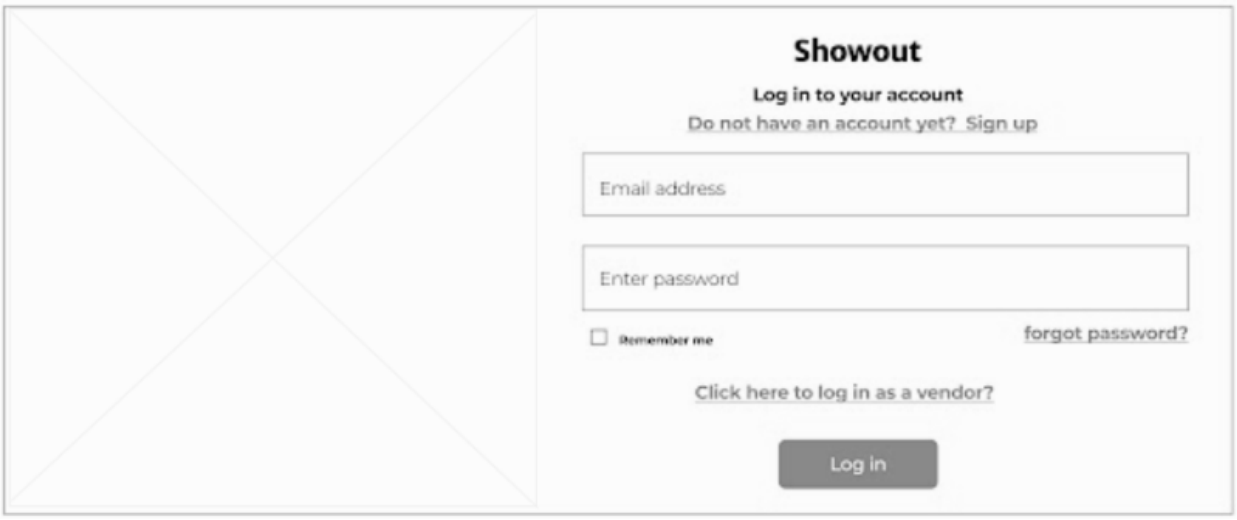
An example of an ecommerce web application similar to what is developed in this project is the Shopify platform. Shopify was built using Software as a Service (SaaS) and it provides a user-friendly interface for both vendors and customers (Main, 2023). SaaS provides users the convenience in setting up their accounts with variety of themes and plugins which inures to the enhancement of the page. However, compared to other frameworks, SaaS have limited customisation compared to frameworks like Django and WordPress plugin. BigCommerce also uses a framework similar to what was adopted in this project. BigCommerce is most used by vendors with high business turnovers. The framework's database has the capacity to accept large volumes of businesses. Again, Amazon Marketplace was built based on the marketplace model which allows vendors to sell their products on the Amazon platform. Comparing these web applications to what is being built in this project, this project introduces a web application that allows vendors to sell their products and as well, for service providers to also render their services to customers.

Software Design

The platform’s user interface is easy to navigate and intuitive, and it is responsive across devices. The application platform is also designed to protect sensitive data from unauthorized access as the organization grows, the platform can easily be expanded to accommodate growth.

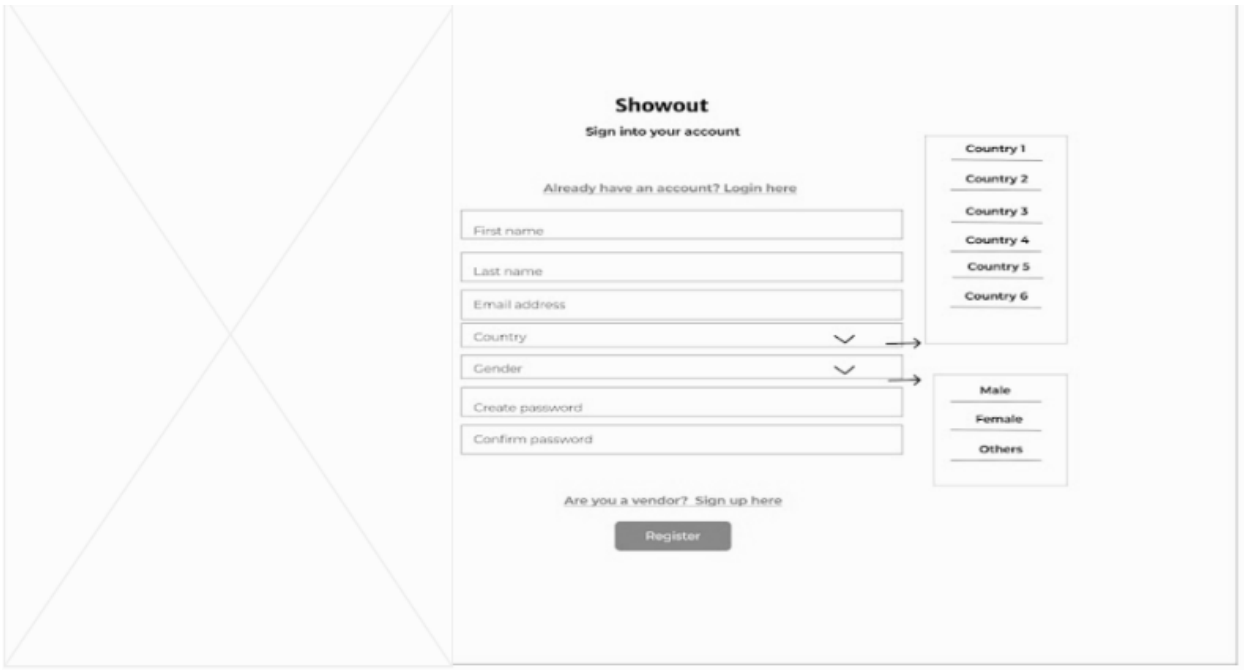
3.1 Wireframes

Customer Login form



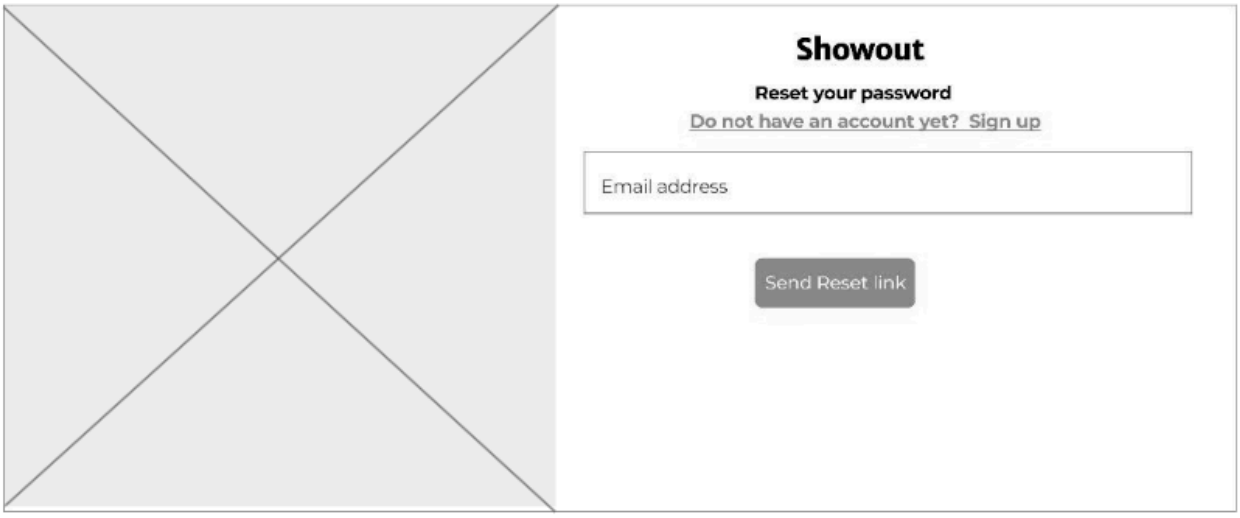
This wireframe for the login section of the platform showcases the login interface for existing customers on the web application.At the top of the page is the logo of the business, other elements include the platform name, login instructions, account creation link, email and password fields, Remember me option, Forgot password link, and a Sign In button. The platform also includes a Don't have an account? Sign up link for new users. The Remember me option recommends preserving login credentials for further visits, whilst the "Forgot password" link facilitates account recovery. There is also a link to the vendor login page.

Customer registration form



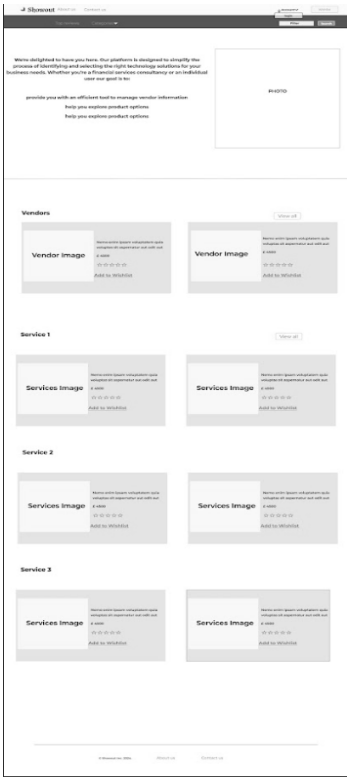
The wireframe above illustrates the customer registration page for the ShowOut platform, including a title, slogan, login redirect, and a form field for entering personal details such as First Name, Last Name, Email, Contact Number,Country, Gender, Password, and Confirm Password. A prominent Sign Up button enables the completion of the registration process after users fill out the form. After a new customer has entered his/her details in the provided inputs field, the user clicks on the register button to register him/her.

Customer Password reset page.



The wireframe above illustrates the customer password reset page for the ShowOut platform, including a title, slogan, sign up redirect, and a form field for entering an Email address and a send reset link button. Reset button sends a reset password link to the customer email after filling out the form. If the customer doesn't have an account they click on the sign up redirect link which takes them to the sign up page .

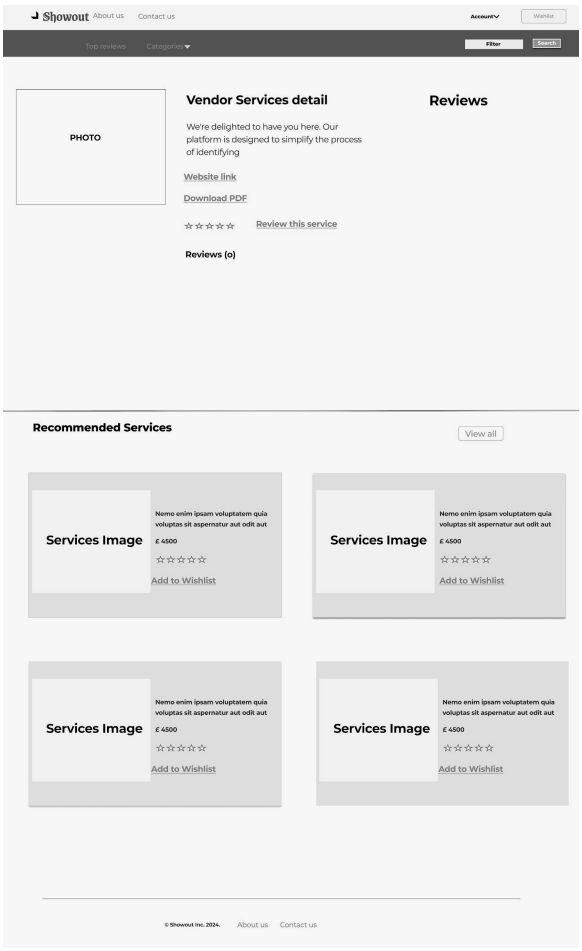
Customer home page



The image is a wireframe of a landing page or homepage for the ShowOut Inventory Platform web application. It includes a navigation bar, the sub navigation bar, search functions, content sections with mission statements and an overview of goals. The navigation bar links to Showout, Contact us, and About us, while the sub-navigation bar allows users to browse vendors based on reviews and categories.

Browse Services link allows a customer to browse through the categories of services rendered by the web application. The request button presents the number of services a customer has added to their request list. The request history button takes a customer to page where he/she sees all the requests they have made on the web application. The account button gives a customer the chance to make changes to his personal account that he/she is using to make request on the page. The changes include changing password, name, etc. Finally, the logout button is the button users use in logging out from their respective accounts.

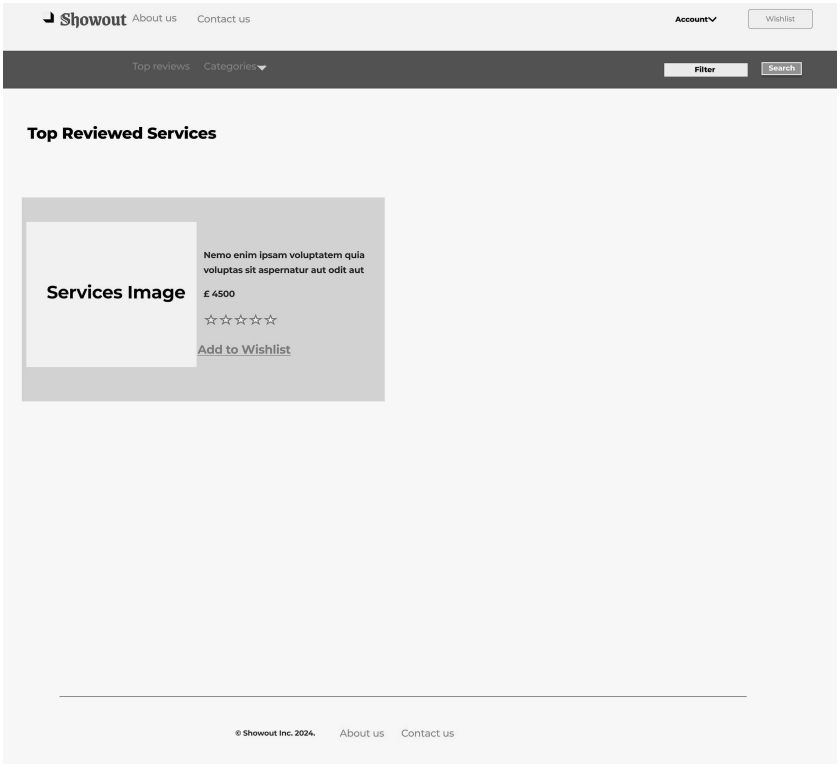
Service page



The wireframe displays a section of the ShowOut platform dedicated to services, showcasing essential elements such as a search bar, content area, and interactive features like the Rate this service functionality. The popup or modal includes tabs for both Review and Rating, accompanied by an interactive star rating system, allowing users to submit their feedback. This implies a user-friendly mechanism for leaving reviews and ratings on services or vendors.

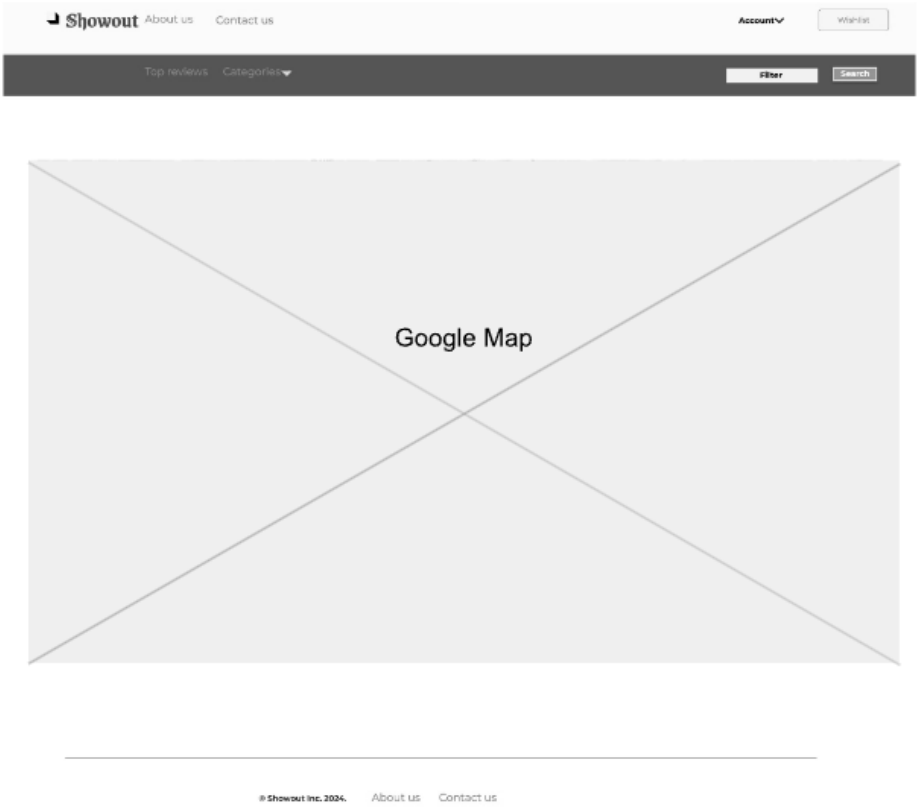
The service details page also has a logo and page title at the top. The page has the service name, description of the service and reviews of the service. The page also has a button for request services and another sections to show recommendations to the customer base on what they like etc

Top reviewed service page



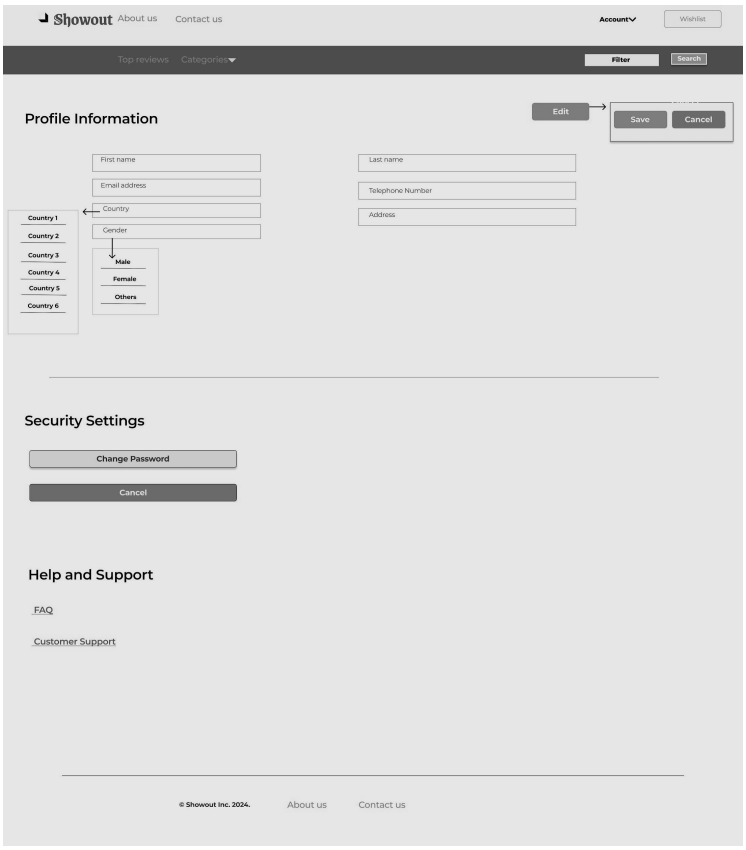
The wireframe displays a section of the ShowOut platform dedicated to top reviewed services, showcasing essential elements such as a search bar, content area, and interactive features like add to wishlist functionality. This page loads the service which is most reviewed on the platform which helps users to be able to get easy access to quality services.

Contact Us page



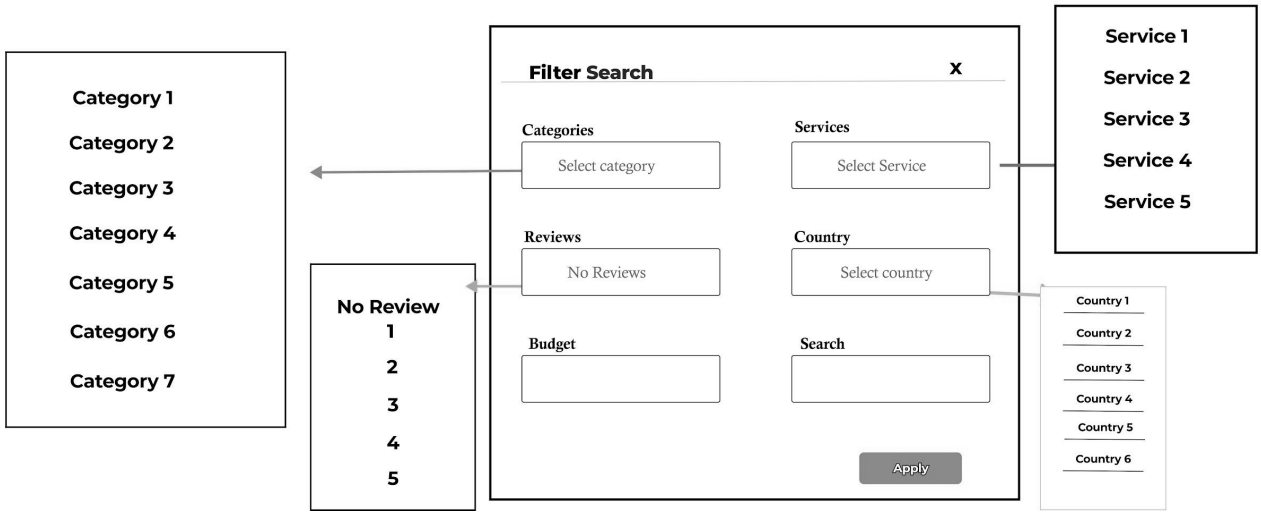
The wireframe displays a section of the ShowOut platform dedicated to the contact us page, showcasing the location of the company displaying on google map. It helps customers and vendors to easily reach out to the Admin for enquiries

Customer Settings Page



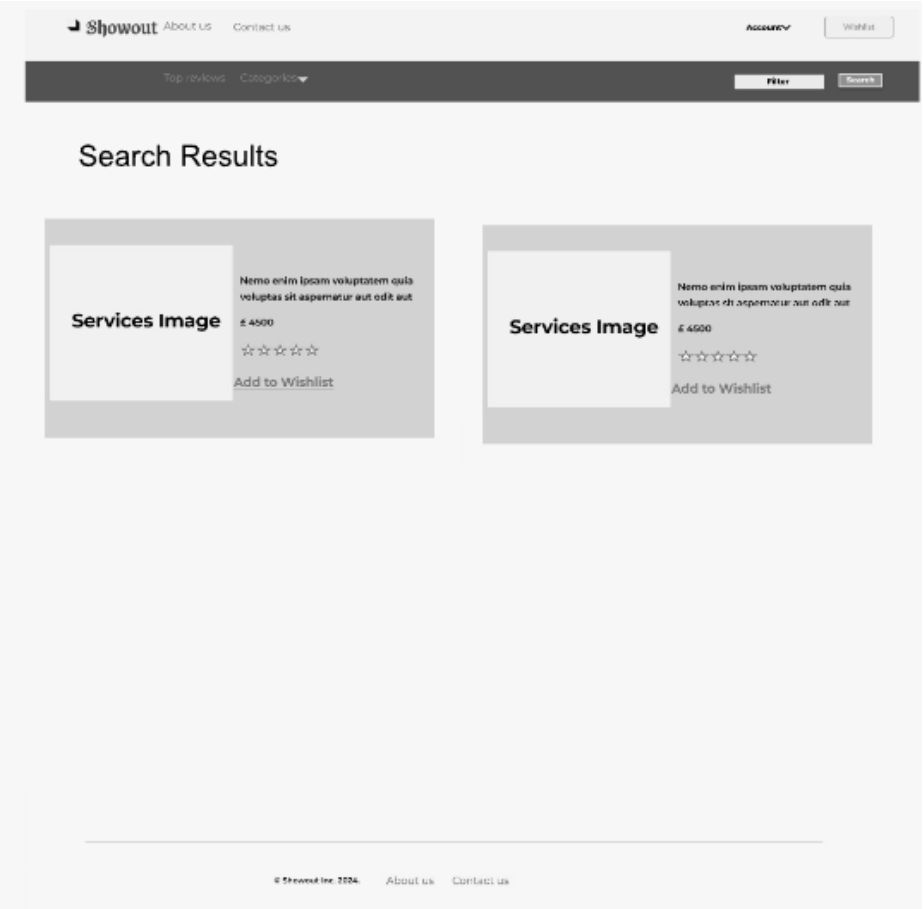
The above wireframe displays the page title at the top of the page. The page has five input fields which are first name, last name, email, contact number, address respectively and 2 dropdowns which are country and gender. The page has a button for saving changes. There's also a security settings section which has a button for changing password.

Search Filter Modal



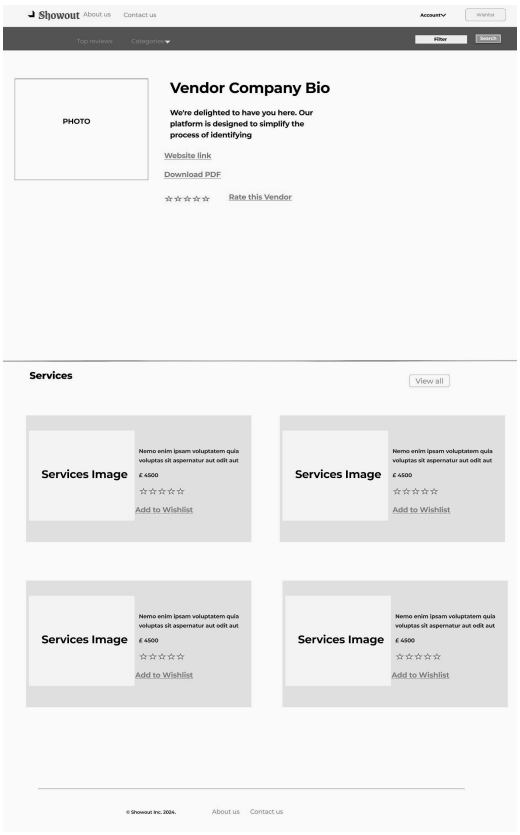
The above wireframe displays the Search Filter Modal with title displayed at the top. The customer can search by filtering by category, service, reviews, country, budget/cost of service and a search query with a button called apply. This search filter helps customers to narrow their search and also makes their search more specific and targeted .

Search Result Page



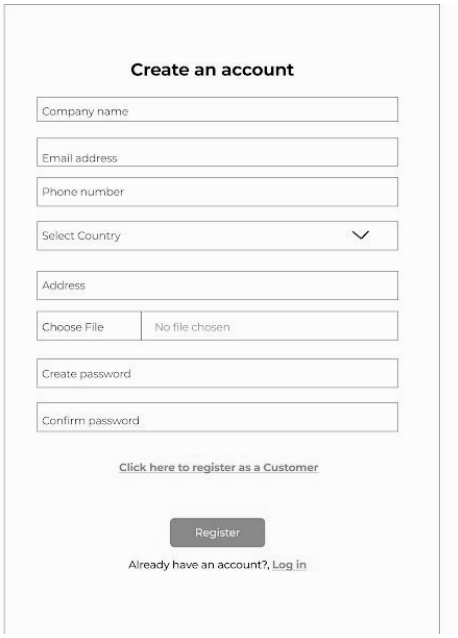
The above wireframe displays the Search Results page with the logo and page title displayed at the top. The search query entered is also displayed at the top of the page beneath the title of the page. The search results are displayed in a vertical format below the search query. The search result page displays only the services based on the search filter modal.

About Vendor Page



The wireframe represents a section of the platform dedicated to the vendor page. Key elements include a top navigation bar with links, a secondary navigation bar for browsing, a vendor section featuring details about a vendor, a star rating system for vendors based on their services, a services section displaying services offered by the vendor.. The presence of a search bar suggests that customers can search for specific vendors or services within the platform.

Vendors’ registration form

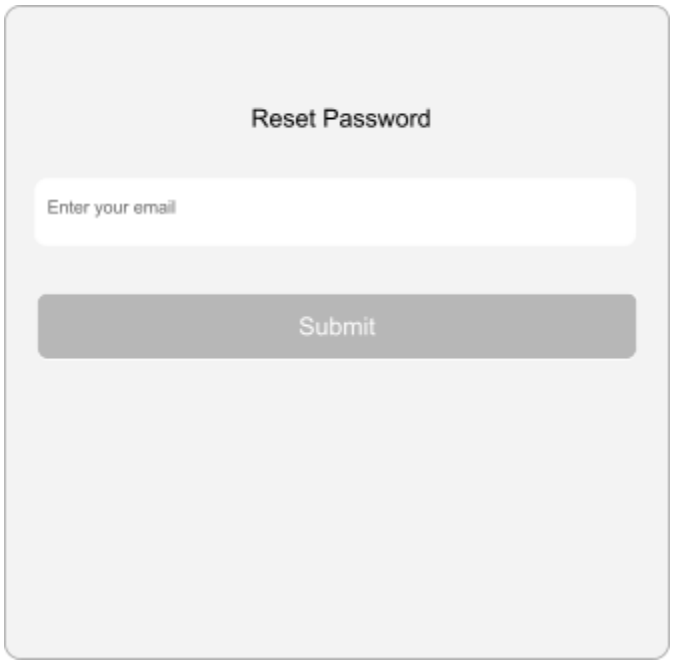


The wireframe above illustrates the vendor registration page for the ShowOut platform, including a title, login redirect, and a form field for entering personal details such as First Name, Last Name, Email, Contact Number,Country, vendor image, Password, and Confirm Password. A prominent Sign Up button enables the completion of the registration process after users fill out the form. After a new vendor has entered his/her details in the provided inputs field, the user clicks on the register button to register him/her.

Vendors Login form

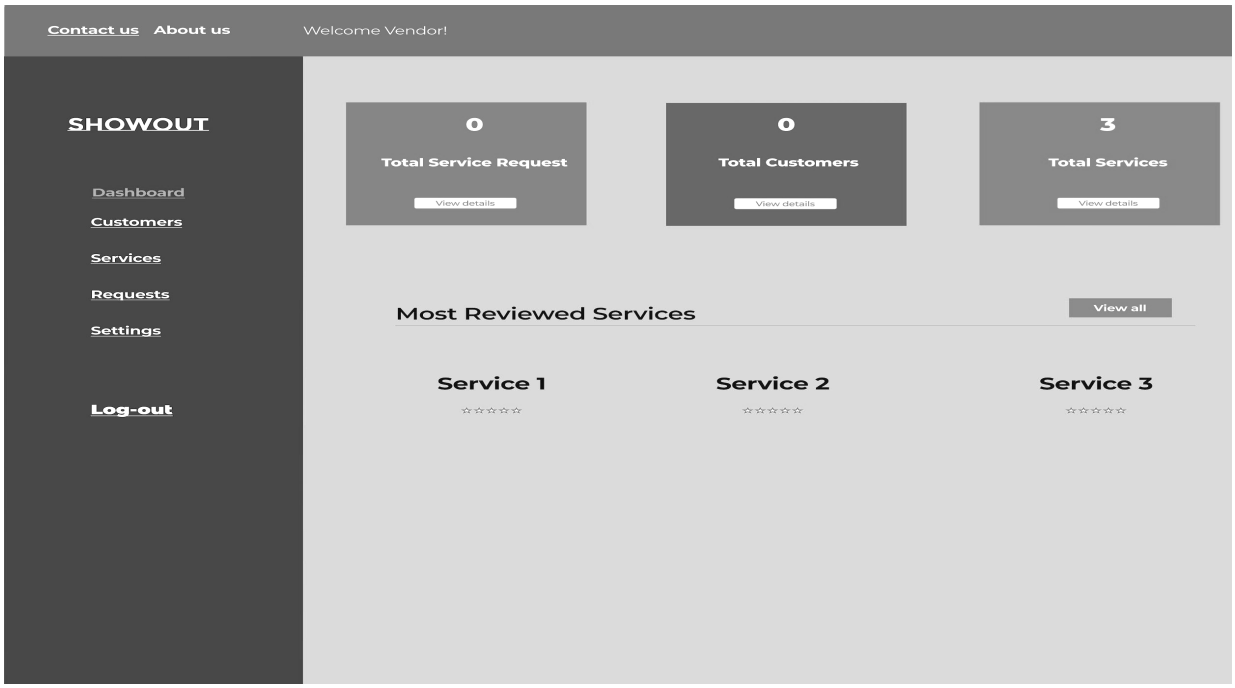
This wireframe for the login section of the platform showcases the login interface for existing vendors on the web application. At the top of the page is the platform name, login instructions, account creation link, email and password fields, Remember me option, Forgot password link, and a Sign In button. The platform also includes a Don't have an account? Sign up link for new users. The Remember me option recommends preserving login credentials for further visits, whilst the "Forgot password" link facilitates account recovery. There is also a link to the vendor login page.

Vendors Reset Password Page



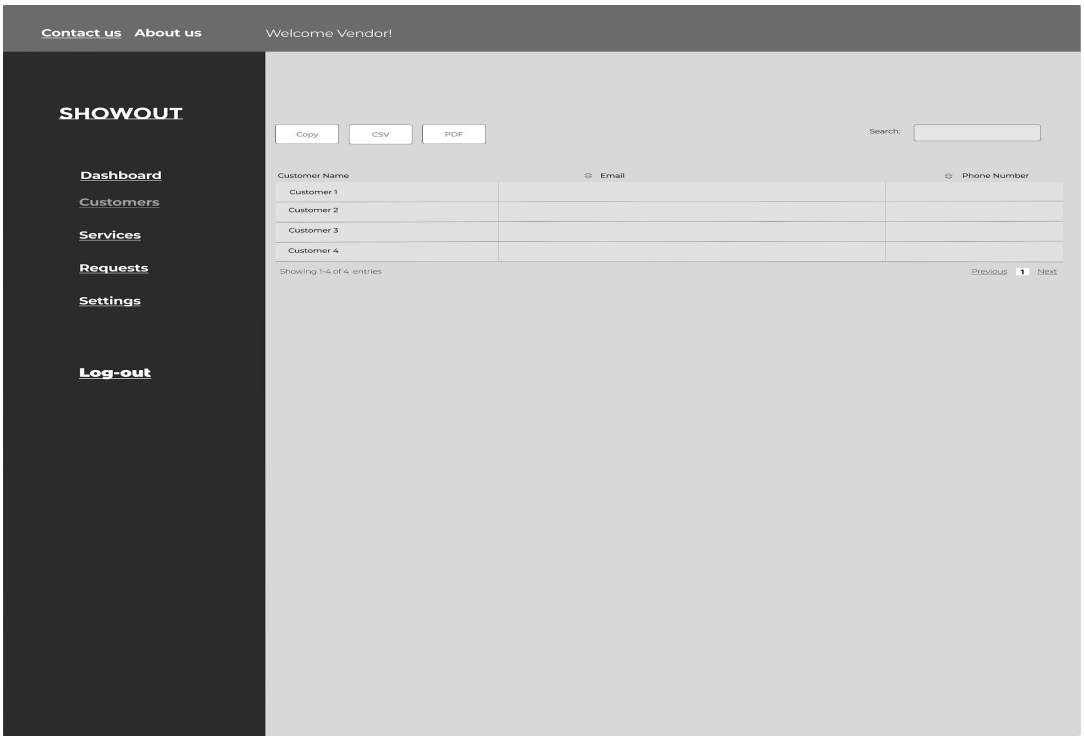
The wireframe above illustrates the vendor password reset page for the ShowOut platform, including a title and a form field for entering an Email address and a submit button. Submit button sends a reset password link to the vendor's email after filling out the form. If the vendor doesn't have an account they will be redirected to the sign up page .

Vendors' Dashboard



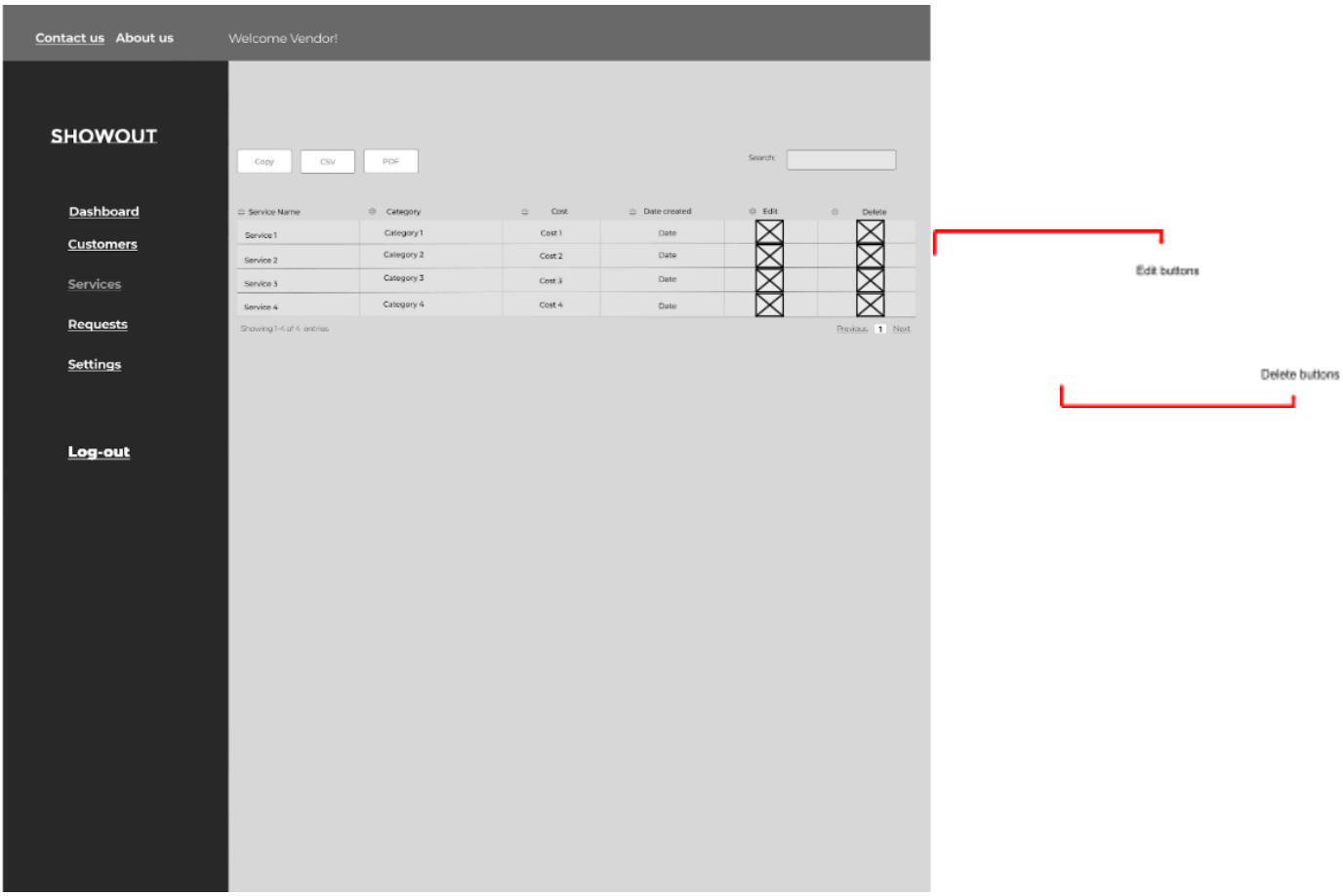
The vendor dashboard page is the page the vendor is redirected to when they have been authenticated either through registration or login process. The vendor dashboard page is divided into 3 sections, the navbar, Sidebar and the content page. The navbar has contact us, about us and a welcome message to the vendor.. The Sidebar has a list of links such as Customers link which redirect the users to the page that shows all the customers who have sent a request to the vendor, the service link redirect the use to their list of services they have added, three request links takes them to the list of request received, the settings link takes the vendor to their account setting page where they can update their profile details, finally thee logout link logs the user out by destroying the user session which takes the vendor to the login page. The content page with the cards shows the count of customers, services and requests belonging to the authenticated vendor, below it shows their most reviewed service.

Vendors' Customers Page



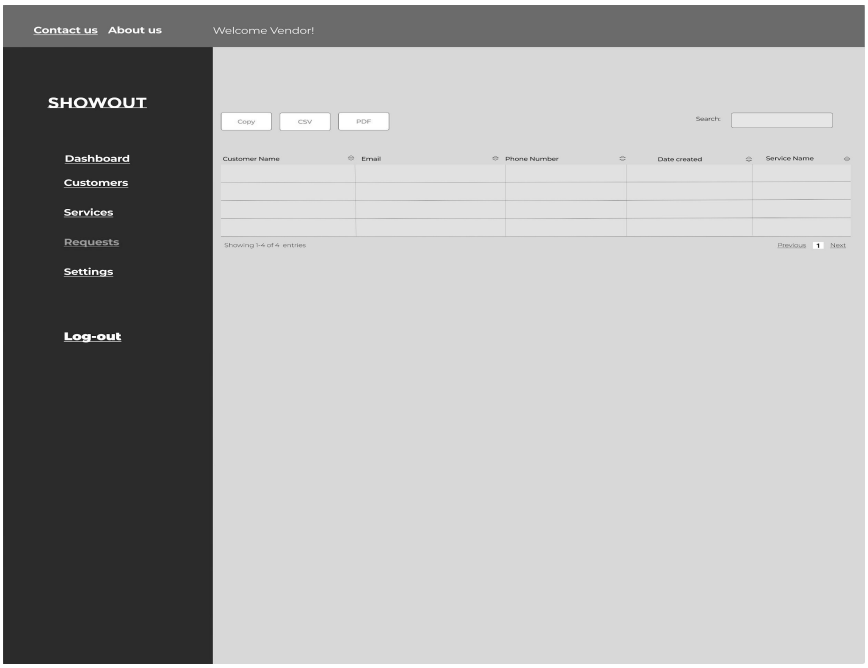
The vendor customer page shows all the customers who have sent a request to the vendor, it loads the important information about each customer like customer full name, email and phone number in a table form. It also has three buttons that help the vendor to export the data in a csv, pdf form or copy it to clipboard. It also has search functionality which helps in filtering or searching for a specific customer. Finally it also has a pagination to show the vendor the total number of pages and content.

Vendors' Services Page



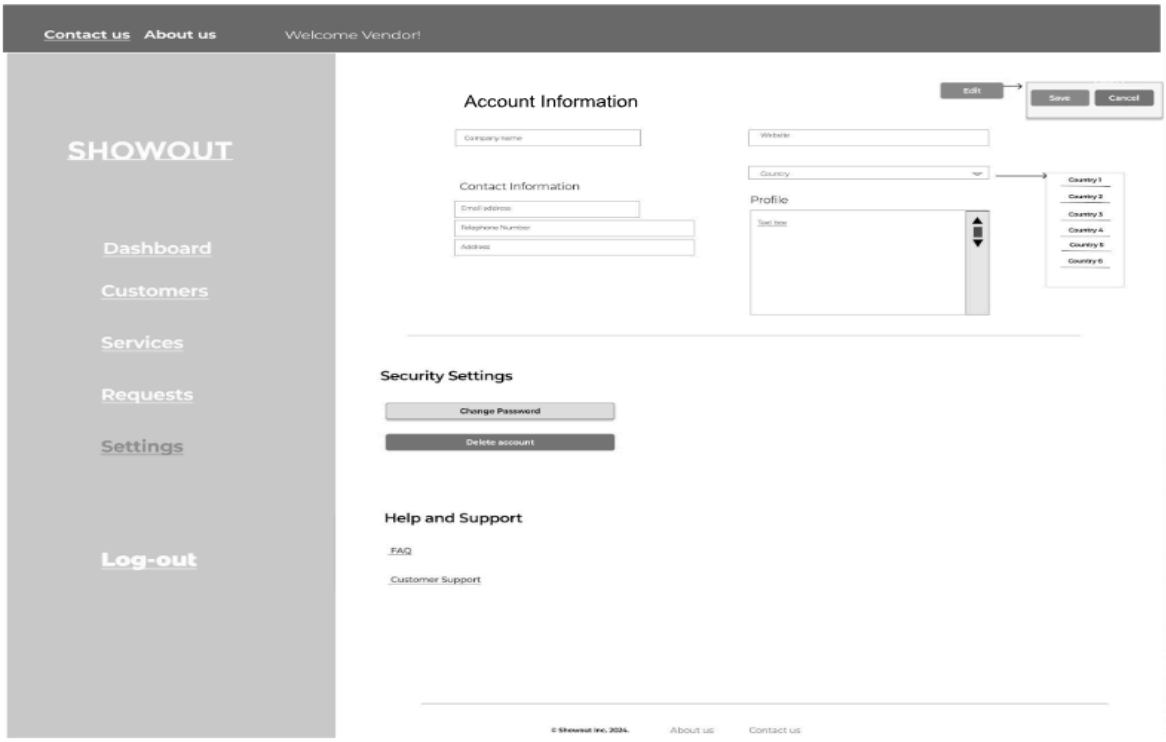
The vendor service page shows all the services the vendor has added to the platform, it loads the summary of the service data like service name, category, cost, and date the service was added or updated in a table form. It also has three buttons that help the vendor to export the data in a csv, pdf form or copy it to clipboard. It also has search functionality which helps in filtering or searching for a specific service. The page also has an edit and delete buttons, the edit takes the vendor to the edit service page and the delete button deletes the service and also has a add service button that allows the vendors to add services, Finally it also has a pagination to show the vendor the total number of pages and content.

Vendors' Customer Requests Page



The vendor customer request page shows all the requests from customers who have sent a request to the vendor, it loads the important information about each customer and the service like customer full name, email and phone number and service name in a table form. It also has three buttons that help the vendor to export the data in a csv, pdf form or copy it to clipboard. It also has search functionality which helps in filtering or searching for a specific customer or service. Finally it also has a pagination to show the vendor the total number of pages and content.

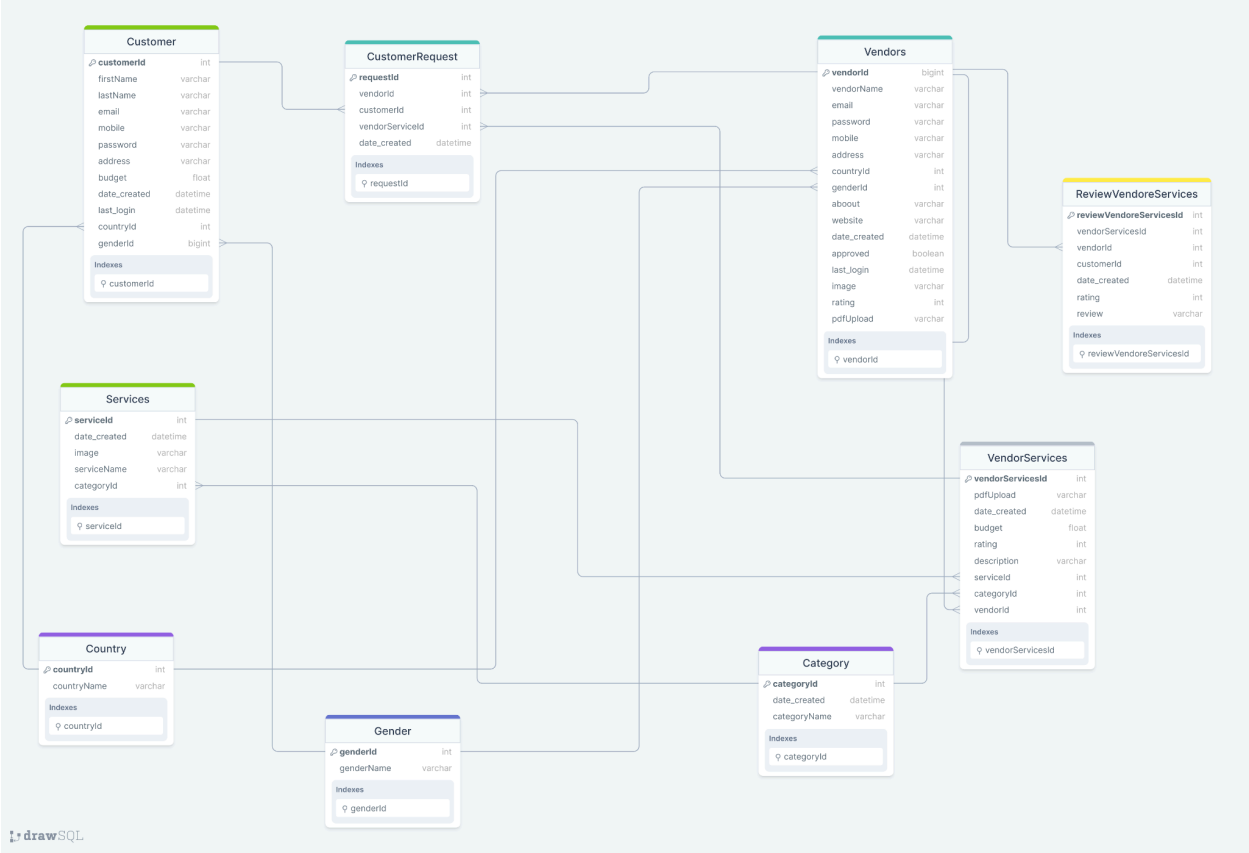
Vendors' Settings Page



The above wireframe displays the page title at the top of the page. The page has five input fields which are company name, email, contact number, address respectively and 1 dropdown which is the country and profile/about the company. The page has a button for saving changes. There's also a security settings section which has a button for changing password. Also has a delete button in case the vendor wants to delete the account.

3.2 Database Design

The database is structured to support a service-oriented platform, consisting of various tables in a form Class Models in django that store information about countries, customers, vendors, categories, services, vendor services, and customer request, gender, Reviews. Below is the detailed schema for each table and the relationships between them:



Several forms of relationships were established between the tables. The relationship between Customers and CustomerRequest is one-to-many indicating that one customer can have multiple CustomerRequest associated with it. The customer table serves as the core and central entity point representing all customers of the websites. According to Fernigrini (2021), a

one-to-many relationship is established by having a primary key in one table that corresponds to a foreign key in another table.

Again, the relationship between vendors and vendor services is also one-to-many. This relationship allows vendors to register as vendors and manage their services on the website.. The relationship between vendors and services is important in the management of service listings and requests. In this one-to-many relationship, each vendor can have multiple services associated with them. Johnston (2014) intimated that one-to-many relationships are common in relational databases, where one entity is associated with multiple instances of another entity. This relationship supports vendors in diversifying their offerings beyond physical products. The relationship between customers and requests is fundamental for managing shopping cart functionality on the website. The relationship allows each customer to have multiple items in their shopping cart.

The relationship between Customers and Customer Requests helps in order management and tracking for customers on the website. This is also a one-to-many relationship where customers can request for multiple services over time.

The many-to-many relationship between Customers and Services is enabled through the Request table which acts as a link table linking customers to the Service they have requested. The relationship makes it possible for many customers to request many services and vice versa. This relationship enables the website to accurately track and manage customer requests, including the services requested. Many-to-many relationships are commonly implemented using junction tables to represent complicated associations between entities (Ramakrishnan and Gehrke, 2003).

The database relationships outlined are important in facilitating the functionalities of the website for both vendors and customers. From managing user accounts and service listings to request and account settings, these relationships ensure interactions and data integrity within the web application. The implementation of these relationships in the database schema enables the website to effectively serve the needs of its users and provide an all-inclusive e-commerce experience.

Country Table

The foundation of the schema is the country table, which stores a list of countries, each uniquely identified by a countryId. This table is essential for assigning geographic locations to both customers and vendors, thereby enabling location-based service offerings and searches.

```
-- Country Table
CREATE TABLE `Country`(
   `countryId` INT NOT NULL,
   `countryName` VARCHAR(255) NOT NULL,
   `countryCode` VARCHAR(255) NOT NULL,
   PRIMARY KEY(`countryId`)
);
```

Customer Table

Moving on to the platform's main users, the customer table contains all pertinent data about them, such as their name, contact information, and a link to their user account via the user_id field. Customers' nationalities are linked to the country_id column, which provides essential data for location-specific service offerings. The customer table is essential for customising the platform's customer experience.

```sql
-- Customer Table
CREATE TABLE `Customer`(
   `customerId` INT NOT NULL,
   `firstName` VARCHAR(255) NOT NULL,
   `lastName` VARCHAR(255) NOT NULL,
   `email` VARCHAR(255) NOT NULL,
   `mobile` VARCHAR(255) NOT NULL,
   `password` VARCHAR(255) NOT NULL,
   `address` VARCHAR(255) NOT NULL,
   `budget` DOUBLE(8, 2) NOT NULL,
   `date_created` DATETIME NOT NULL,
   `last_login` DATETIME NOT NULL,
   `countryId` INT NOT NULL,
   `genderId` BIGINT NOT NULL,
   PRIMARY KEY(`customerId`)
);
```

Vendors Table

Parallel to the customer table is the vendors table, which details the service providers on the platform. It has fields for the names of the vendors, their contact details, service details, and their location via a country_id reference. In order to provide a reliable marketplace, the approved field is a boolean that indicates if a vendor has been checked and given permission to provide services.

```sql
-- Vendors Table
CREATE TABLE `Vendors`(
   `vendorId` BIGINT NOT NULL,
   `vendorName` VARCHAR(255) NOT NULL,
   `email` VARCHAR(255) NOT NULL,
   `password` VARCHAR(255) NOT NULL,
   `mobile` VARCHAR(255) NOT NULL,
   `address` VARCHAR(255) NOT NULL,
   `countryId` INT NOT NULL,
```

```
    `genderId` INT NOT NULL,
    `aboout` VARCHAR(255) NOT NULL,
    `website` VARCHAR(255) NOT NULL,
    `date_created` DATETIME NOT NULL,
    `approved` TINYINT(1) NOT NULL,
    `last_login` DATETIME NOT NULL,
    `image` VARCHAR(255) NOT NULL,
    `rating` INT NOT NULL,
    `pdfUpload` VARCHAR(255) NOT NULL,
    PRIMARY KEY(`vendorId`)
);
```

Category Table

The platform's navigability and user-friendliness are improved by arranging services properly into categories that are maintained within the category table. Each service listed in the services table is linked to a category through the category_id field, ensuring that services can be easily filtered and found by customers seeking specific types of service offerings.

```
-- Category Table
CREATE TABLE `Category`(
    `categoryId` INT NOT NULL,
    `date_created` DATETIME NOT NULL,
    `categoryName` VARCHAR(255) NOT NULL,
    PRIMARY KEY(`categoryId`)
);
```

VendorServices Table

The vendorservices table functions as a many-to-many relational table that ties vendors to the services they offer, categorised under category. This junction table is the cornerstone of the platform's service listing functionality, enabling vendors to list multiple services and customers to view a variety of offerings from each vendor.

```sql
-- VendorServices Table
CREATE TABLE `VendorServices`(
   `vendorServicesId` INT NOT NULL,
   `pdfUpload` VARCHAR(255) NOT NULL,
   `date_created` DATETIME NOT NULL,
   `budget` DOUBLE(8, 2) NOT NULL,
   `rating` INT NOT NULL,
   `description` VARCHAR(255) NOT NULL,
   `serviceId` INT NOT NULL,
   `categoryId` INT NOT NULL,
   `vendorId` INT NOT NULL,
   PRIMARY KEY(`vendorServicesId`)
);
```

Request Table

Lastly, the wishlist table records the preferences and interests of customers. It acts as a ledger of services that customers are interested in, linking them to both the services they wish to procure and the vendors who offer these services. The wishlist lets users bookmark and revisit their favourite services, enabling a personalised user experience.

```sql
-- CustomerRequest Table
CREATE TABLE `CustomerRequest`(
   `requestId` INT NOT NULL,
   `vendorId` INT NOT NULL,
   `customerId` INT NOT NULL,
   `vendorServiceId` INT NOT NULL,
   `date_created` DATETIME NOT NULL,
   PRIMARY KEY(`requestId`)
);
```

ReviewVendorServices Table

Lastly, the review vendor services table records the reviews of the customers. It acts as a ledger of services that customers are interested in or not, linking them to both the services they wish or have already procure and the vendors who offer these services. The reeviews also help other customers to make good financial decisions and helps other vendors to improve on their services. This table contains the vendorId, customerId and serviceId as foreign key and a primary key as reviewVendorServicesId, with review/comments and the rating value.

```
-- ReviewVendorServices Table
CREATE TABLE `ReviewVendoreServices`(
   `reviewVendoreServicesId` INT NOT NULL,
   `vendorServicesId` INT NOT NULL,
   `vendorId` INT NOT NULL,
   `customerId` INT NOT NULL,
   `date_created` DATETIME NOT NULL,
   `rating` INT NOT NULL,
   `review` VARCHAR(255) NOT NULL,
   PRIMARY KEY(`reviewVendoreServicesId`)
);
```

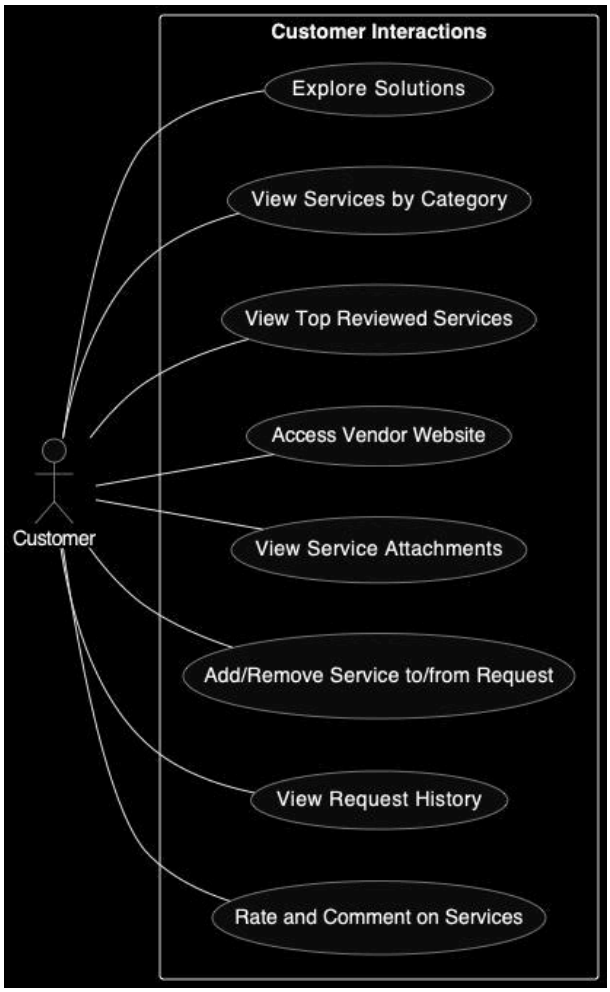
Gender Table

Lastly, the gender table helps the admin to populate the list of genders so that it will show on the frontend as a dropdown to avoid users entering wrong names etc. The gender table contains the gender name and genderID as the primary key, which will be use as a foreign key in other table

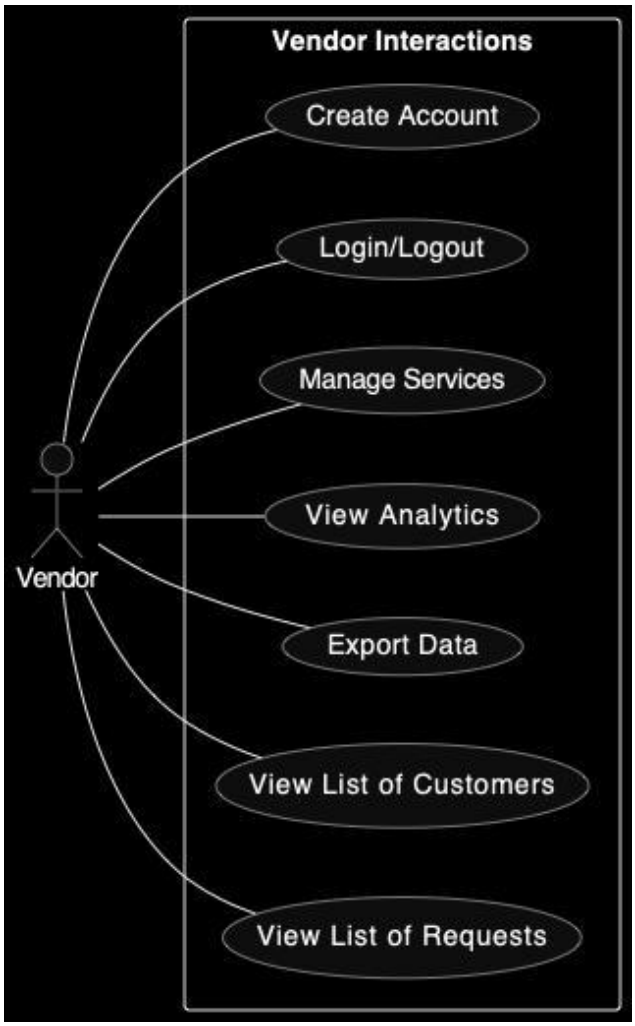
```
-- Gender Table
CREATE TABLE `Gender`(
   `genderId` INT NOT NULL,
   `genderName` VARCHAR(255) NOT NULL,
   PRIMARY KEY(`genderId`)
);
```

3.3 Software Design

Customer Use Case Diagram

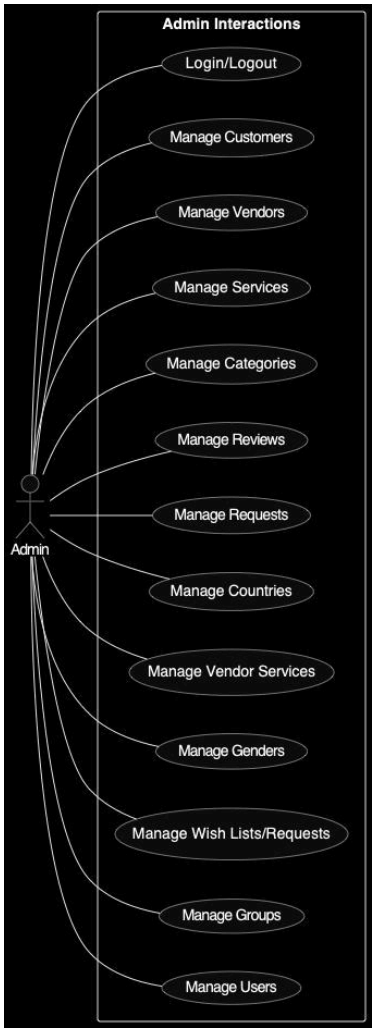


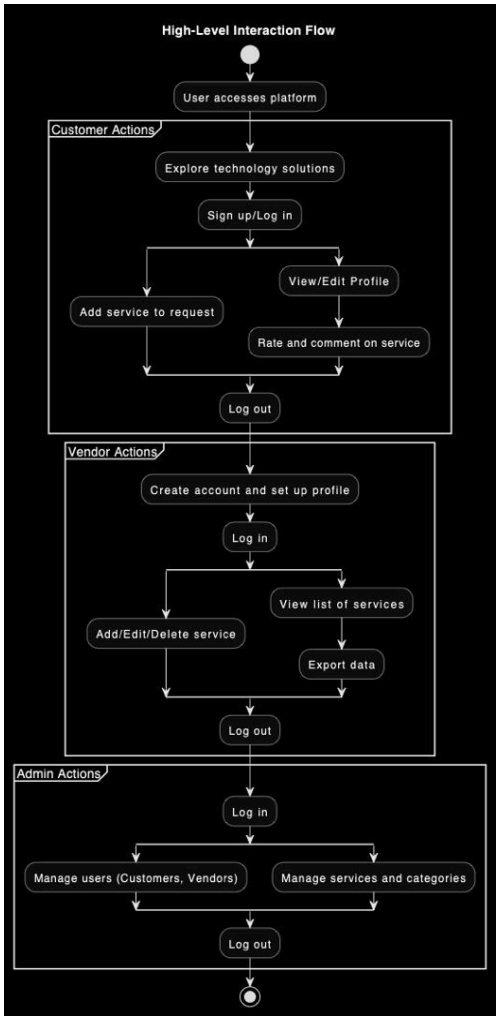
Vendor Use Case Diagram



Admin Use Case Diagram

High Level Flowchart





Customer Use Case Diagram

This diagram demonstrates the interactions of a customer with a Vendor Application Inventory Platform. Specifically, a customer has several options to find the necessary solutions; they can pick the ones platform offers, considering there is a great variety of them by types: that is, the customer can make the list or sort of solutions available and take a selection, It is also possible for a customer to find out whether there any customers rated it, read the reviews i.e. top-reviewed services feature. Additionally, a platform allows branching to the vendor’s website and sends it to an enterprise's available source for a customer to seek more about the vendor or

collaborate with it through its digital facilities. Also, in order to get a full picture, a customer can read the attached datalliance, such as detailed service descriptions, instructional materials, or media content.

Further, since the platform is interactive as well, the customers should be allowed to add or remove services to request lists, perhaps modeled as a shopping cart, which can then be used for inquiries or even as a stepping stone to procurement. Similarly, the customers should be able to view their request histories, which can serve as a valuable record of interactions made, services mused over, or transactions completed on the platform in the past. Third, the platform allows for a community feedback system, wherein the customers can rate and comment on the services provided to them, thereby creating a repository of user-generated insights, guiding other customers in the future. In this way, the feedback aspect is marginalized as well.

Vendor Use Case Diagram

The use case diagram of the opportunity identifies a brief business requirement to access vendors in a platform environment created to meet their demands. Initially, the vendor must create an account before establishing themselves in the platform. This element provides a baseline requirement that allows a vendor to personalize their space on the platform. After that, the vendor can log in and out at will, ensuring that no unauthorized person accesses their account, hence data. The Vendor shall update, remove, and add their services per market conditions in a feature function known services.

Apart from service management, the platform provides the view of analytics, which is important for the vendors to analyze their customers, service provision performance, and business decision-making. Another critical feature provided by the platform is data export function, which

allows the vendors to download their business data and utilize it accordingly, for example, deeper analysis or presenting it to relevant audiences. Lastly, the vendors can view a full list of customers. This is vital for customer relationship management and helps in marketing and selling.

Finally, the system enables vendors to review a list of requests from customers, such as inquiries about services and support issues. The mentioned functionality ensures that vendors respond to customer needs timely and effectively. They also help keep their satisfaction levels high and implement independent operations management. In sum, the system diagram above is illustrated to be a robust tool that provides vendors with the necessary instruments to provide their services, communicate with their clients, and conduct operational analysis through the system.

Admin Use Case Diagram

The use case diagram for the Admin Interactions in a platform depicts a broad range of administrative duties. For starters, the admin is allowed to log in and out of the system in a secure way, ensuring that the platform's administrative functionalities are not easily accessible. Upon successful authentication, the admin has the power to manage customer profiles, handling their requests and monitoring their interaction with the platform. Additionally, it entails managing the vendors, which comprises monitoring their profiles and activities, authoring prospective vendors, and ensuring delivery of customer service until pa is invoiced.

The admin is in charge of the services available on the platform. They have the mandate to ensure that all the listings are current while at the same time making sure all entries are properly grouped . The admin is also critical in filtering reviews. This is through reviewing and approving or rejecting the feedback from users.

Another primary admin function is responding to various requests, including inquiries from customers or purchasing or servicing orders submitted by users. Admins also carry out global operations, such as managing country-specific developments in the platform to better meet the requirements of various regions.

The admin also manages the specific services provided by each vendor, ensuring vendor listings are up-to-date and appropriate to the platform's standards. The admin should also manage demographic data, including genders users may select in their profiles, and ensure the platform remains an inclusive and respectful place for shorthand of user identity. The admin also manages wish lists and service requests, including tracking and responding to the things and services users wish for. Also if the platform has any community-oriented features, the admin manages any groups, including creating groups, approving membership in groups, and setting group access controls.

Lastly, the admin‘s role includes complete user supervision such as new users onboarding and updating or disabling user accounts . Consequently, the administrator guarantees that the user idealism on the platform is flawless and that the user is at the heart of their activities. Therefore, the administrator is instrumental in the proper running of the platform, overseeing and regulating it to ensure that the platform is serving the best interest of all its stakeholders and delivering the best service and user experience.

High level Flowchart

The use case diagram describes how various interactions in a platform will flow, separating the different paths taken for the system by customers, vendors, and administrators . For the customer, the interaction with the platform begins by investigating various technology innovations. They then proceed to create new accounts or log into their existing accounts. Once they have their accounts open, the customer sets their preferences, including their profile information. Outwardly, they also have the option to add the service to the request list after which they would wish to make an inquiry or a purchase, as well as rate and comment after evaluating the service. Finally, the customer logs out after their time on the platform.

The vendors will engage with the platform by creating and setting up their profiles. The latter step will introduce their services to the platform's ecosystem. Vendors' capabilities also cover other operations such as adding, editing, or deleting their service listings. Such functionality will ensure that their offerings remain up-to-date and responsive to the evolving market needs. Meanwhile, the key to vendors' interaction will be the view and export of data concerning their services. Export functionality will ensure that data can be stored in alternative systems for business analysis or record-keeping purposes. Vendors will complete their sessions by logging out, securing the completion of their administrative activities.

The role of the administrators is also structured, whatcomming in the form of utilizing a secure login to access the pool of management tools offered to run effective supervision. They manage the user accounts by adding customers and vendors, and even manage the services and category sections. Consequently, the platform stays in a position to look organized and usable, and the services listed are primary and fit. With the administrative part balanced, the administrator logs out, marking an ending to the management. The user administrator section summarizes the roles and connections that ensure a seamless affair within the platform.

5. Testing

8 major tests were conducted for this application with each test representing a different case. All the tests have been written and shown in the table below. These tests were conducted for the following features, i.e. register, login, reset password, all vendors, homepage, services, rate services and request.

NO.	Feature	Scenario	Expected Outcome	Output	Result	Chrome	Firefox	Edge	Safari	Opera
1.	Login	Correct Username & Password	Successful Login	Login Successful	Pass	√	√	√	√	√
2.		Wrong username and password	Error	Incorrect information	Pass	√	√	√	√	√
3.		Wrong username and correct password	Error	Incorrect Username	Pass	√	√	√	√	√
4.		Correct username and wrong password	Error	Incorrect Password	Pass	√	√	√	√	√
5.		Empty username and password	Request for input	Field must be filled	Pass	√	√	√	√	√
6.		Correct username and empty password	Request for input	Field must be filled	Pass	√	√	√	√	√
7.		Empty username and password	Request for input	Field must be filled	Pass	√	√	√	√	√
8.	Register	Valid username and password	Registration Successful	Successfully Registered	Pass	√	√	√	√	√
9.		Already existing user	Already Registered User	Already Registered User	Pass	√	√	√	√	√

10.		Invalid username and valid password	Error	Check username	Pass	√	√	√	√	√
11.		Invalid username and password	Error	User not found	Pass	√	√	√	√	√
12.		Valid username and invalid password	Error	Weak Password	Pass	√	√	√	√	√
13.		Username and empty password	Request for input	Field must be filled	Pass	√	√	√	√	√
14.		Empty username and password	Request for input	Field must be filled	Pass	√	√	√	√	√
15.		Username, password and empty email address	Request for input	Field must be filled	Pass	√	√	√	√	√
16.		All empty	Request for input	Filled must be filled	Pass	√	√	√	√	√
17.	Reset Password	Valid email address for a registered user	Email sent	Email Delivered	Pass	√	√	√	√	√
18.		Valid email address for unregistered user	Error	Mail not delivered	Pass	√	√	√	√	√
19.		Invalid email address for a registered user	Error	Mail not delivered	Pass	√	√	√	√	√
20.		Invalid email address for unregistered user	Error	Mail not delivered	Pass	√	√	√	√	√
21.	All Vendors	Selecting Top Reviews	Leads you to another page	Displays the Top Reviews page	Pass	√	√	√	√	√
22.		Click on Browse Categories	More option displays	Several categories of services are displayed	Pass	√	√	√	√	√
23.		Enter a query in the search and click on same	Operations opens another page	Displays the search page showing	Pass	√	√	√	√	√

				search results						
24.		Select one of the vendors	Leads to another page	Opens the page of the vendor displaying the products and services rendered by the vendor	Pass	√	√	√	√	√
25.		Click on request	Redirect to another page	Request page is displayed	Pass	√	√	√	√	√
26.		Click on Account Settings	Redirect to a different page	Displays the accountings page	Pass	√	√	√	√	√
27.	Homepage	Click on About Us	Leads to a different page	Displays a page that shows information about the portal	Pass	√	√	√	√	√
28.	Services	Select a service from the list of recommended services	The page is redirected to a different one	Displays details of the selected services	Pass	√	√	√	√	√
29.	Rate Services	Rate the service and submit	Success	Rated Successfully	Pass	√	√	√	√	√
30.		Write a comment and submit	Success	Rated Successfully	Pass	√	√	√	√	√
31.	Request	Place a request to a vendor	Request being processed	Success!	Pass	√	√	√	√	√
32.		Select request history	Opens another page	Displays history of previously made requests	Pass	√	√	√	√	√

6. Conclusion

The ShowOut Inventory Platform is a solution to the challenges faced by financial services consultancies. The platform features a dynamic Vendor Application Inventory Platform that centralises a marketplace for vendors to showcase their services and users to explore, evaluate, and engage with offerings. Rigorous testing prioritises the development process, with a user-friendly interface, advanced search capabilities, detailed vendor profiles, and a centralised data repository. ShowOut aims to revolutionise the financial services industry by driving efficiency, informed decision-making, and enhanced profitability. It presents a dependable resolution to the complex obstacles encountered by financial services experts in the constantly progressing technological environment.

This project has been a learning curve for me and has significantly sharpened my coding skills. At the end of the project, I developed an ecommerce web application using the Django framework. This web application provides a one stop shop for customers to place requests for services. There are a variety of services and products that a customer could select from. The application also leveraged some of the advanced security features in Django to enhance the security of the web application. However, more security features could be integrated into the application to enhance its security features. Future studies may also consider the possible integration of the web application to a financial technology (fintech) service provider.

REFERENCES

Omotunde, H. and Ahmed, M., 2023. A Comprehensive Review of Security Measures in Database Systems: Assessing Authentication, Access Control, and Beyond. Mesopotamian Journal of CyberSecurity, 2023, pp.115-133.

Pärn, E.A., Edwards, D.J. and Sing, M.C., 2017. The building information modelling trajectory in facilities management: A review. Automation in construction, 75, pp.45-55.

Zhu, F. and Liu, Q., 2018. Competing with complementors: An empirical look at Amazon. com. Strategic management journal, 39(10), pp.2618-2642.

Jarvis, R. and Han, H., 2021. Fintech Innovation: Review and Future Research Directions. International Journal of Banking, Finance and Insurance Technologies, 1(1), pp.79-102.

Adegbaju, E. and Odun-Ayo, I., (Oct 2020). Development of a Cloud-Based Inventory Management System.

Forte, A. (2016). *BUILDING A MODERN WEB APPLICATION USING AN MVC FRAMEWORK*. [online] Available at: https://core.ac.uk/download/pdf/38141681.pdf.

García, S., Rueda, S., Florián Gaviria, B. and Bacca Cortés, B. (2017). Programmable mobile robots as hands-on platform for basic programming. *Sistemas y Telemática*, 15(41), pp.57–73. doi:https://doi.org/10.18046/syt.v15i41.2455.

Haleem, A., Javaid, M., Qadri, M.A. and Suman, R. (2022). Understanding the Role of Digital Technologies in education: a Review. *Sustainable Operations and Computers*, [online] 3(3), pp.275–285. doi:https://doi.org/10.1016/j.susoc.2022.05.004.

Hoang, T.V., Van, H.A.T., Hoang, H.T., Vo, H.T. and Chansomphou, V. (2023). Acute hepatitis in a patient with multiple biliary hamartomas. *Oxford Medical Case Reports*, [online] 2023(9). doi:https://doi.org/10.1093/omcr/omad072.

Kelton (2021). *Django vs Flask: Picking the Right Python Web Framework | kellton*. [online] www.kellton.com. Available at: https://www.kellton.com/kellton-tech-blog/django-vs-flask-picking-the-right-python-framework.

Main, K. (2023). *What Is Shopify And How Does It Work (2023 Guide)*. [online] Forbes Advisor INDIA. Available at: https://www.forbes.com/advisor/in/business/software/what-is-shopify/.

Martin-Rios, C., Erhardt, N.L. and Manev, I.M. (2022). Interfirm collaboration for knowledge resources interaction among small innovative firms. *Journal of Business Research*, [online] 153, pp.206–215. doi:https://doi.org/10.1016/j.jbusres.2022.08.024.

Rabl, T., Sadoghi, M., Jacobsen, H.-A., Gómez-Villamor, S., Muntés-Mulero, V. and Mankowskii, S. (2012). Solving Big Data Challenges for Enterprise Application Performance Management. *arXiv:1208.4167 [cs]*. [online] Available at: https://arxiv.org/abs/1208.4167.

Salas-Zárate, M. del P., Alor-Hernández , G., Valencia-García, R., Rodríguez-Mazahua, L., Rodríguez-González, A. and Cuadrado, J.L.L. (2015). Analyzing best practices on Web development frameworks: The lift approach. *Science of Computer Programming*, [online] 102, pp.1–19. doi:https://doi.org/10.1016/j.scico.2014.12.004.

Tadhani, J.R., Vekariya, V., Sorathiya, V., Alshathri, S. and El-Shafai, W. (2024). Securing web applications against XSS and SQLi attacks using a novel deep learning approach. *Scientific Reports*, [online] 14(1), pp.1–17. doi:https://doi.org/10.1038/s41598-023-48845-4.

Verma, P. (2023). *Unlocking Performance: A Guide to Async Support in Django*. [online] DEV Community. Available at: https://dev.to/pragativerma18/unlocking-performance-a-guide-to-async-support-in-django-2jdj.