

MVP Vinyls: Project Report

Friday, April 23rd, 2021



Presented by **The Real MVPs:**

Name	Student ID	Email	Role
Angela Yao	z5115096	a.yao@student.unsw.edu.au	Developer
Angus Feng	z5207356	angus.feng@student.unsw.edu.au	Developer
Benjamin Pick	z5162064	b.pick@student.unsw.edu.au	Maintainer
Mandy Tao	z5060806	m.tao@student.unsw.edu.au	Developer
Michael Gysel	z5251938	m.gysel@student.unsw.edu.au	Scrum Master

Table of Contents

Background	3
Overview	4
System Architecture	4
External Actors	5
Presentation Layer	5
Business Layer	5
API Layer	6
Database Layer	6
Objectives and Functionalities	8
Objective 1: Account Management	8
Objective 2: Browsing Vinyl Records	8
Objective 3: Recommendations	8
Objective 4: Inventory Management	9
Objective 5: Shopping Cart	9
Objective 6: Payments	9
Objective 7: Chatbot	10
Objective 8: Business Insights	10
Third-Party Functionalities	11
Frontend	11
Backend	13
Implementation Challenges	17
Recommendation Algorithm Choices	17
MongoDB Relational Connections	18
Image handling	18
Sharing API Wrapper instances	19
User Documentation/Manual	20
Configuration	20
Setup	20
Usage and Functionalities	21
References	47

Background

Problem

Anyone who has ever purchased vinyl records online has struggled finding new records they actually want to buy. There are two main causes of this: e-commerce vinyl record websites hinder the user's ability to easily browse records and offer ill-suited record recommendations.

E-Commerce companies selling vinyl records rarely specialise in selling vinyl records; moreover, this lack of specialisation leads to difficult record browsing and ill-suited record recommendations. When entering Amazon, the largest retailer of new vinyl in the world, users are not shown recommendations for vinyl records to purchase, but for new Prime Video shows to watch, ceramic bowls to buy, and foot cream (Forbes, 2018; RIAA, 2018). In order to search for a vinyl record, the user must type in a keyword, select 'CDs & Vinyl' under a long dropdown list, press search, where they are then directed to search results filled with CD's and Cassettes and hopefully vinyl records. There is no browsing vinyl records on Amazon.

E-Commerce companies that do focus on selling vinyl records allow users to browse vinyl records to purchase, but offer ill-suited record recommendations. When searching 'David Bowie & Morrissey - Cosmic Dancer' on Discogs, the largest retailer of used vinyl records in the world, the user is only shown recommendations of other David Bowie records or other Morrissey records (Discogs, 2020; RIAA, 2018; Discogs, 2021). Amazingly, the user is shown recommendations for three different versions of Cosmic Dancer, the record they just searched for. Clearly, these recommendations do not allow users to discover the new records they desire.

Solution

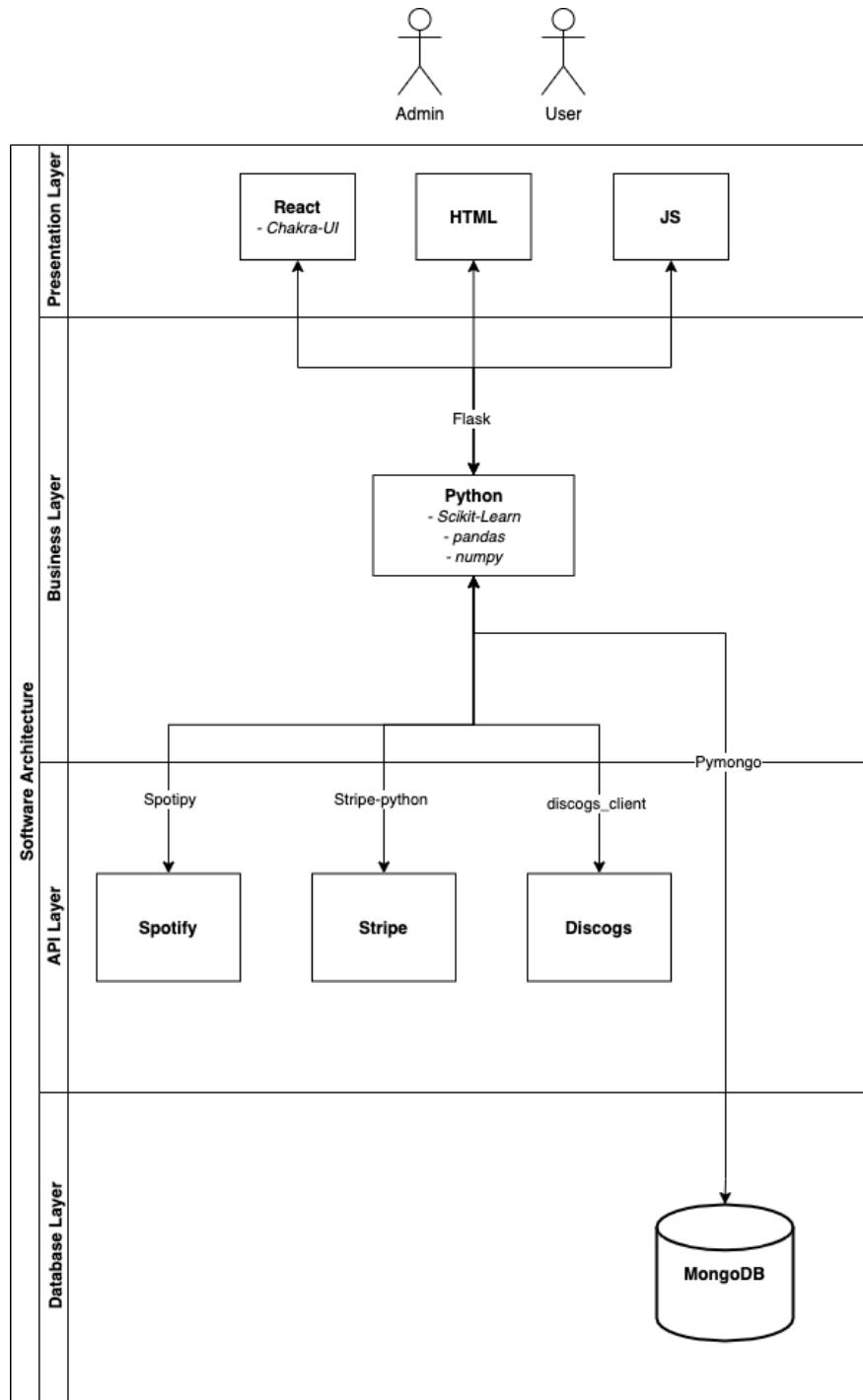
MVP Vinyls is an e-commerce vinyl record company that empowers users to find records they actually want by solving these browsing and recommendation issues in the existing marketplace.

Users can login to their accounts; browse for vinyl records to purchase; receive record recommendations based on a specific vinyl record, based on their order history, and based on their Spotify listening history; get any questions answered by a chatbot; and of course, purchase the records themselves. Notably, these recommendations leverage the wealth of digital music data through the Spotify API in order to refine product recommendations for each user. As more music is listened to through streaming services than any other individual source, MVP Vinyls' recommendation system is unparalleled in the existing marketplace (RIAA, 2018). Lastly, administrators can login to their accounts where they can manage inventory and view business insights from sales data, giving them all the tools necessary to maximise profits for MVP Vinyls.

Overview

Architecture/design of the overall system and functionalities

System Architecture



Our choice of system architecture is designed modularly such that our two main teams (backend and frontend) can work towards completing features independently.

External Actors

There are two main external actors: the user and administrator. Both interact through the presentation layer as an admin does not always have the technical skills to make direct database requests. Admins have the same functionality as users, such that they can view all the vinyl records and receive recommendations; however, admins also have access to an admin panel, which allows for actions such as promoting existing users to admins, managing the inventory, and viewing sales and order data.

Presentation Layer

The presentation layer allows for the user and admin to interact with our system through a user interface. We used React JS as our main driver for frontend development, abstracting HTML and JS into components (Reactjs, 2021). React allowed for the use of reusable components, a feature not found in vanilla JS or HTML, which provided both a consistent looking user interface and modular frontend code. To assist us, we used Chakra UI as our UI component library which provided ready-made UI components for our React app (Chakra-UI, 2021). This allowed for an increase in development speed, provided an attractive look and also improved the accessibility of the website. Similarly, React's dynamic rendering of these components allowed for complex JSON objects to be displayed generically, such as ones produced for recommendations and product pages, relying less on hardcoding implementations. We chose MongoDB as our database, which supplies JSON records and allowed for native rendering in React with little processing (MongoDB, 2021). Here, the Python Flask framework provides REST API endpoints for React to send requests to, to either fetch, update, delete or give data to our Python backend (Flask, 2021).

Business Layer

The business layer provides both a connection between the frontend and all other components, but also allows for the processing and storing of data. This layer is backed by Python, chosen due to the need for heuristics in the backend. Scikit-learn drives how we solve recommendations, providing out-of-the box solutions such as classification algorithms and n-dimension cosine similarity, commonly used in recommender systems (Scikit-learn 2021). In order to use this algorithm, however, we must pre-process our Spotify information into a vector, where numpy and pandas can be used to format and create structured DataFrames that Scikit-learn can digest (Pandas 2021, Numpy 2021). The use of Python allowed us to combine both the Flask endpoint implementation and all other API and processing packages; moreover, this allows for a singular server instance to host our backend if the client chooses to deploy this product, such as Google Cloud Platform (Google Cloud Platform 2021). On the other hand, AWS provides easy to implement lambda functions in Python through the use of virtualenv, if the client chooses

to in the future (Amazon Web Services, 2021). Although Python provides powerful packages such as Requests for creating API requests, our chosen APIs provide abstracted wrappers, namely Spotipy, discogs_client and stripe-python, simplifying a lot of complex procedures such as authorisation into function calls (Spotipy, 2021; Discogs_client, 2021; Stripe-python, 2021). To interact with our database layer, we used Pymongo which allows for inserting, updating and modifying records by inputting Python objects, thereby reducing the complexity between converting backend data structures to database records (Pymongo, 2021).

API Layer

The API layer provides interaction with external services that implement predefined solutions. The music-based APIs we used are Spotify and Discogs. Spotify provides a trusted way of obtaining track, album and artist statistics, which was the foundation of our recommender algorithm. Spotify also provides authorisation through which we created personalised recommendation services based on a user's listening history, liked songs, and playlists. Discogs provided us with mock data for populating our vinyl inventory system; however, it was realistic enough to link Spotify to the music representing the vinyl records, allowing for accurate recommendations to be made. Both Spotify and Discogs use OAuth 2.0, allowing for both secure storage of data as well as informing the user of how the web application is restricted to use their data through the use of scopes (OAuth 2.0, 2021).

MVP Vinyls considered a multitude of payment APIs including Stripe, Square, PayPal and numerous lesser known payment APIs such as Adyen and FIS Global. Because of the six week development deadline, we narrowed the search to Stripe, Square, and Paypal due to their depth of developer support. Because MVP Vinyls has global aspirations, we excluded the Square API as it only accepts credit card payments in five countries (Square, 2021). Lastly, because the Stripe API is significantly cheaper than the PayPal API in every country it operates, Stripe was chosen as the payment API (Stripe, 2021; PayPal, 2021).

Database Layer

The database layer allows for persistence of both user and administrator data, such as their cart and order history, as well as item data, such as vinyl records to sell and recommendation processing. MongoDB, being a NoSQL document database, focuses on storing JSON objects over storing data in fixed columns and rows. This allowed for seamless passing from the database to the business layer to the presentation layer. This also provides scalability for the client, whereas large inventories of vinyls are created with numerous users and orders, the system remains stable and responsive. MongoDB provided an interface for iterating over large sets of items, known as a cursor, where limits and offsets can be provided, which allowed for quick and precise processing of only what is necessary to the frontend, such as when viewing multiple pages of items

(MongoDB, 2021). This, combined with the use of indexing provides users with a combination of reliability and responsiveness when viewing large datasets.

Objectives and Functionalities

Objective 1: Account Management

The Account Management objective allows users to enjoy a personalised e-commerce experience and administrators to manage product inventory and obtain business insights from sales data.

In order to achieve this objective, MVP Vinyls implemented the ability for both users and administrators to sign up for an account, to login, to logout, and to reset their password. In addition, users and administrators can edit their personal information and shipping address. Lastly, in the admin panel, administrators can promote users to administrators and demote administrators to users. This way, administrators can give coworkers the ability to access all administration features and help with the business workload.

Objective 2: Browsing Vinyl Records

The Browsing Vinyl Records objective allows users to find vinyl records they wish to purchase. MVP Vinyls achieves this objective by giving users the ability to search vinyl records and view vinyl record recommendations.

Once a user logs into their account, they enter their homepage, where they can search for new vinyl records based on artist, song name, or genre. If a user searches for a particular vinyl record, they are shown a catalogue of results, with each result leading to its own product page. Back on the user homepage, the user is also shown a variety of recommendations, as discussed below. Thus, through these functionalities, the user has a multitude of ways to browse new vinyl records to purchase.

Objective 3: Recommendations

The Recommendations objective allows users to more easily discover vinyl records they want to purchase. MVP Vinyls achieves this objective through a novel, multi-pronged approach not being used by other e-commerce companies selling vinyl records.

Upon logging in, users can view vinyl record recommendations based on products they have purchased in the past or based on their Spotify listening history. These recommendations leverage the unprecedented wealth of digital listening data through the Spotify API in order to recommend products based on product similarity, as opposed to products other users purchase. Furthermore, they leverage the user's own digital listening history in order to make recommendations. Next, if a user navigates to a product page, they are also shown recommendations for vinyl records specific to that product.

Through these functionalities, users are consistently shown new vinyl records they may want to purchase.

Objective 4: Inventory Management

The Inventory Management objective allows administrators to effectively manage inventory in order to aid administrators in achieving company profitability.

MVP Vinyls achieves this objective in the admin panel, only accessible to administrators. In the products tab of the admin panel, administrators can view products in the product catalogue and search specific products by album name or artist. Importantly, administrators can view and update the inventory levels for each product. If the administrator wants to add a new product to the catalogue, they can add a product by inputting its album name, artist, quantity, price, genres, songs, and an image. If an administrator wants to edit an existing product or delete discontinued products, the products tab also gives them the ability to do so. Through these functionalities, administrators can effectively manage inventory.

Objective 5: Shopping Cart

The Shopping Cart objective enables users to more easily purchase vinyl records they want by allowing them to add, view, and edit items in their shopping cart.

In order to achieve this objective, MVP Vinyls designed a shopping cart icon that is visible when the user logs in. This way, the user can easily navigate to their shopping cart regardless of their current location. From any product page, the user can add a specific product quantity to the shopping cart. At their shopping cart page, the user can view the total price as well as the price and quantity of each record, allowing them to easily coordinate purchases around their budget. If a user wants to update their shopping cart, they have the ability to edit the quantity or delete any items as well. Lastly, the user's shopping cart is stored in the database; thus, if the user logs out, their previous shopping cart will remain in place. This collection of features enables users to easily decide what vinyl records to purchase.

Objective 6: Payments

The Payments objective allows users to purchase products they want to listen to.

To achieve this objective, MVP Vinyls built a secure checkout and payment portal using the Stripe API. When a user clicks 'Checkout' from the Shopping Cart page, they are redirected to the Checkout page, where the user's stored name and shipping address are automatically entered, and which the user can edit. From here, the user can enter the Checkout page, giving them a summary of their order. From the Checkout page, the user can then enter the Stripe payment gateway, where the user can securely enter their

credit or debit card information and purchase the products in their shopping cart. If the payment succeeds, the user is notified of the successful order where they can redirect to their homepage; however, if the payment fails, the user is redirected to the Checkout summary page. Thus, the user can easily and securely purchase the vinyl records they want to listen to, achieving the Payments objective.

Objective 7: Chatbot

The Chatbot objective enables users to easily find information about both products and orders, so that they can stay informed of products they may want to purchase or products they have already purchased. While existing e-commerce companies specialising in selling vinyl records all provide users with a combination of frequently asked questions and email contact forms, MVP Vinyls provides a novel chatbot that more intuitively answers user questions.

In order to achieve this objective, MVP Vinyls allows logged in users to click on a chatbot icon on their homepage. Upon clicking the chatbot icon, a user can click ‘Product’, ‘Delivery’, or ‘Warranty’. If a user clicks ‘Product’, they can then enter a keyword and are returned relevant products relating to that keyword. Clicking on a product then directs the user to the corresponding product page. If a user clicks ‘Delivery’, they can enter an order number they have placed, where the chatbot will then return the estimated delivery date and a link to the corresponding product page. If a user clicks ‘Warranty’, they can then enter the name of a product and the chatbot will return the warranty for that product as well as a link to the product page itself. Through these functionalities, the chatbot provides a novel and intuitive interface unparalleled in the world of e-commerce vinyl records.

Objective 8: Business Insights

The Business Insights objective allows administrators to view sales and order data to enable them in making informed business decisions that will generate the company profit.

MVP Vinyls achieved this objective in the admin panel, specifically through the Dashboard and Orders tabs. In the Dashboard tab, administrators are met with graphs of revenue over time and new customer orders over time; two indicators critical to keeping the business profitable. Furthermore, the Orders tab then allows administrators to view all orders, including the order number, shipping email, date of order, and the ability to view and fulfill the order. Here, the administrator can also search orders by order number in order to allow them to view specific orders. Through these functionalities, MVP Vinyls has provided administrators with significant tools to aid in running their company.

Third-Party Functionalities

Frontend

React

React is an open source, JavaScript library developed by Facebook used for building front end user interfaces and reusable UI components (Reactjs, 2021). The reason we decided to use React is because it is a frontend library that our developers are comfortable with, it is one of the most popular frontend frameworks and it is a library that is fast and easy to develop with.

React is an open source software licenced as MIT which permits commercial use, modification, and distribution.

Create React App

Create React App is a tool that generates a frontend build pipeline using a pre-configured webpack for development (Reactjs, 2021). The reason we decided to use Create React App is because it is the simplest way to start the development of a react app, it saves us time by removing almost all setup and configuration and it removes the need to maintain a complex build pipeline. For the purpose of this project we didn't require any extensive customisation of the configuration.

Create React App is an open source software licenced as MIT which permits commercial use, modification, and distribution.

React Router / React Router Dom

React Router is a package that allows for handling routes in a web application (React Router, 2021). The package contains a collection of navigational components which allow you to implement a component based approach to routing. MVP vinyls is designed as a predominantly single page application (SPA) using React Router. SPA's don't require page reloading and instead dynamically rewrites the current web page. This allows for a fast and responsive web application.

React Router is an open source software licenced as MIT which permits commercial use, modification, and distribution.

Styled Components

Styled Components is a package that allows for the use of CSS syntax inside JavaScript (Styled Components, 2021). This allows us to style react components without the need for CSS files. It gives us the benefits of JS such as isolation, better developer experience and

manipulating values using JS as well as the benefits of CSS such as reusability, media queries and behaviours like hover and focus.

Styled Components is an open source software licenced as MIT which permits commercial use, modification, and distribution.

Recharts

Recharts is a composable charting library built with React components (Recharts, 2021). The main purpose of this library is to allow users the ability to create charts with ease. As part of our ‘Business Insights’ objective, we wanted to provide administrators with critical information regarding the website’s revenue and customer orders so that they are able to operate MVP Vinyls in the most efficient and profitable manner. We found that the most appropriate way to do this was by displaying the data into graphs and is the primary reason why we utilised this library.

Recharts is an open source software licenced as MIT which permits commercial use, modification, and distribution.

Chakra UI

Chakra UI is a react component library which is a set of prebuilt, reusable React components (Chakra UI, 2021). The reason we used a component library is because:

- It greatly improves the speed of development
- Components follow a theme so the styling of the website is both aesthetic and consistent
- Components are compatible across browsers
- Accessibility is built into the components

Chakra UI styles components using a style prop framework which further improves the speed of development.

Chakra UI is an open source software licenced as MIT which permits commercial use, modification, and distribution.

React Stripe js

React Stripe js is a thin wrapper around stripe elements (Stripe, 2021) which is a JavaScript library used for building payment flows. Stripe elements contain a set of prebuilt UI components to build the checkout flow. (Stripe, 2021). The reason we use React Stripe JS is to allow the frontend to use React components that work in conjunction with the Stripe API backend.

React Stripe js is an open source software licenced as MIT which permits commercial use, modification, and distribution.

React Chatbot Kit

React Chatbot Kit is a package that contains a customizable, prebuilt React chatbot component (React Chatbot Kit, 2021). The reason we decided to use React chatbot kit is because the specification requires a chatbot. Designing a chatbot using our own JSX and CSS in the timeframe we had would not produce a component that would fit the standard we would like. React Chatbot Kit is customizable enough to fit the project's needs.

React Chatbot Kit is an open source software stating that it is licenced as MIT on npmjs.com. However their official Github contains no licence. If a repository has no license, then all rights are reserved and the code is not open source. Therefore this may pose as an issue for the commercial use of MVP Vinyls.

React Icons

React Icons is a library that offers users the ability to easily include popular icons into their React applications (React-Icons, 2021). We utilised these icons when creating certain buttons to increase usability and enhance the user experience through the matching between system and the real world heuristic. For instance, the user profile and shopping cart icon clearly communicates to the user that clicking on them allows the user to view their profile and cart respectively.

Font Awesome Icons is an open source icons library licenced as CC by 4.0 license. It permits commercial use, modification, and distribution.

Bootstrap Icons is an open source icons library licenced as MIT which permits commercial use, modification, and distribution.

Prettier

Prettier is an opinionated code formatter. It ensures all code conforms to a consistent style by removing the original styling and formatting to the code style enforced by prettier (Prettier, 2021). It also allows custom configuration of the formatting. The reason we decided to use prettier is to enforce a consistent style between multiple members of the team working on the frontend. It also improved development speed by having clean code.

Prettier is an open source software licenced as MIT which permits commercial use, modification, and distribution.

Backend

Flask

Flask is a popular web framework used for developing web applications in Python (Flask, 2021). We decided to use Flask because it is a framework that our developers have

worked with before and has a url routing feature which is particularly suitable for the RESTful interface design of our project.

Flask is licenced under the BSD 3-Clause License, which permits commercial use, modification, and distribution.

Werkzeug

Werkzeug is a WSGI toolkit which is used by Flask (Werkzeug, 2021). We also used the werkzeug.security module for storing and handling user passwords securely.

Werkzeug is licenced under the BSD 3-Clause License, which permits commercial use, modification, and distribution.

Flask-CORS

Flask-CORS is a Flask extension for handling CORS to allow cross-origin AJAX (Flask-Cors, 2021). This allows the frontend application to interact with the backend API.

Flask-CORS is licenced under the MIT License, which permits commercial use, modification, and distribution.

Flask-Mail

Flask-Mail is a Flask extension which provides a simple interface to set up SMTP with a Flask application and send emails (Flask-Mail, 2021). This allows us to send emails to users, such as for password reset codes.

Flask-Mail is licenced under the BSD 3-Clause License, which permits commercial use, modification, and distribution.

MongoDB

MongoDB is a NoSQL, document-based, distributed database (MongoDB, 2021). We decided to use MongoDB as the database for this project because the JSON-like documents translated naturally not only into the object-oriented structure of our backend Python application but also into the JSON objects passed between front and backend required by the RESTful design of our API.

MongoDB is licenced under the Server Side Public License, which permits commercial use.

PyMongo

PyMongo is a Python driver for MongoDB (PyMongo, 2021). It contains tools used for working with MongoDB from Python, that enabled us to more quickly access the database.

PyMongo is licenced under the Apache License 2.0, which permits commercial use, modification, and distribution.

DNSPython

DNSPython is a DNS toolkit for Python (DNSPython, 2021). DNSPython was used as it is a PyMongo requirement.

DNSPython is licenced under the ISC License, which permits commercial use, modification, and distribution.

Numpy

Numpy is a Python package that allows for powerful array computing (NumPy, 2021). Numpy was used in order to sort and reshape arrays throughout the product recommendations process.

Numpy is licenced under the BSD License, which permits commercial use, modification, and distribution.

Python3 Discogs Client

Python3 Discogs Client enables users to query the Discogs database and find information on artists, releases, labels, among other information (Joalla, 2021). Python3 Discogs Client was used to generate the products used throughout the website.

Python3 Discogs Client is licensed to Discogs. Any redistribution of the source code must reproduce the corresponding Copyright notice.

Spotipy

Spotipy is a lightweight Python library for the Spotify Web API (Spotipy, 2021). Spotipy was used in order to make product recommendations based on the similarity of each product, leveraging the wealth of digital listening data through Spotipy.

Spotipy is licenced under the MIT License, which permits commercial use, modification, and distribution.

PyJWT

PyJWT is a Python library that allows the encoding and decoding of JSON Web Tokens (JWT) (PyJWT, 2021). PyJWT was used to generate JWT's for users upon registration or login, in order to keep the users logged into the website.

PyJWT is licenced under the MIT License, which permits commercial use, modification, and distribution.

Pandas

Pandas is a Python package that provides powerful data structures for data analysis, time series, and statistics (Pandas, 2021). Pandas was used to create data structures in the product recommendation process.

Pandas is licenced under the BSD License, which permits commercial use, modification, and distribution.

Scikit-learn

Scikit-learn is a set of Python modules used for machine learning and data mining (Scikit-learn, 2021). Scikit-learn's cosine_similarity module was used to produce product recommendations specific to each user.

Scikit-learn is licenced under the BSD License, which permits commercial use, modification, and distribution.

Stripe

Stripe is a Python library for Stripe's API (Stripe, 2021). Stripe was used to create checkout sessions and return them to the frontend whenever a user was ready to checkout and pay for vinyl records.

Stripe is licenced under the MIT License, which permits commercial use, modification, and distribution.

Jellyfish

Jellyfish is a Python library for doing approximate and phonetic matching of strings (Jellyfish, 2021). Jellyfish was used in the Chatbot in order to match user queries with relevant responses.

Jellyfish is licenced under the BSD License, which permits commercial use, modification, and distribution.

Virtualenv

Virtualenv is a tool to create isolated Python environments that solves dependency and versioning issues (Virtualenv, 2021). We utilised virtualenv to ensure the team was developing using consistent python library dependencies and versions.

Virtualenv is licenced under the MIT License, which permits commercial use, modification, and distribution.

Implementation Challenges

Recommendation Algorithm Choices

When we started the project, we were faced with the dilemma of choosing between the two main types of recommender algorithms: content-based, focusing on using features of users and products to suggest similar products, and collaborative-based methods, using user-item interactions such as time spent viewing certain products for product recommendations. The content-based method was favoured due to the importance of the genre classification produced by Spotify, which can be easily manipulated in Python, which we believe is one of the strongest indicators of classifying a type of listener. Further, collaborative-based methods require time to create a "user-item interaction matrix" through lots of training with real users, which the short project length did not permit, suffering from a problem known as "cold start" (Recommender Systems, 2020).

The main algorithm used to create an efficient product-product recommender system (i.e. viewing a vinyl page and receiving recommendations based on that vinyl) is cosine similarity (Cosine Similarity, 2012). This is done by representing our data as vectors and computing the angle between those vectors when projected in any dimensional space. Cosine similarity is commonly used in comparing the meaning of sentences, where an algorithm such as TF-IDF can be used to provide a numeric representation of how relevant a word is in a corpus, and then applying cosine similarity to the top TF-IDF scores to obtain a score between 0 and 1 of how similar these two sentences are (TF-IDF, 2019). This can be adapted to our use case, using a count of genres as a numeric representation of a certain vinyl, obtained through the Spotify API, and applying the cosine similarity algorithm to all other songs to find the top 5 songs to recommend based on this. The algorithm choice here does not suffer "cold start" and can work on any number of products, however, as the inventory expands, so will the accuracy of recommendations.

However, there was a need to incorporate user interaction in a content-based system, where having only product based recommendations did not suffice for a website which contends large vinyl websites such as Discogs. We solve this by using two user-item interactions, which has little overhead on the system, being Spotify recommendations based on the genres of the top artist's that the users listen to, and recommendations based on the genres of the users' orders. Both solutions use the same cosine similarity algorithm, however, we can now compress all genres of both orders and top artists by creating a vector with each dimension representing a certain genre and the size of the vector representing the count of how many products a user ordered which has that certain genre or a count of how many Spotify artists create music according to that certain genre. These methods scale with both the amount of orders or Spotify artists you listen to, where genres that are consistently relevant to a user will have a vector that

trumps others as its size increases, as well as the amount of products in the inventory. Further, this style of recommendations through user-item interactions allows for suggestions that a user might not yet have been interested in by simply sorting the cosine similarity value in ascending order, to retrieve the "least" recommended vinyls.

MongoDB Relational Connections

We decided to use a cloud based NoSQL solution, being MongoDB, to host all data including product, order and user information, mainly due to the rapid development nature of the MVP Vinyls, where the data type was constantly changing, hence there was no need for ACID compliance. Using this type of database does have its cost however, where it is not meant for creating relational links between collections, as objects can be embedded in complex JSON documents. There were use cases where these relational links were necessary such as each order containing a product and user id, as both products and users must exist outside of orders, which is why there is a need for multiple collections. Initially, whenever an order was loaded, the database was queried every time for each product to get the products details. However, this resulted in very slow times, especially for bulk searching of orders such as in the admin dashboard, as an extreme amount of database calls were being made. There were two solutions to this problem using MongoDB's in-built \$lookup aggregation or caching the results for all relevant products into a dictionary. The latter seemed more beneficial as the MongoDB instance we used provided limited memory and computational power compared to our local machines. This set the standard for all of our bulk database calls where we could minimise the amount of database calls and rely on Python to cache our data between calls. Note that this is purely a decision and this may have been different if we ran MongoDB on the same server which Python was running on. Further, MongoDB provides the ability to use indexes to speed up queries when sorting, where indexes were placed on properties such as album name, vinyl price and date to create fast database queries even for large datasets, at the expense of a small memory cost in the MongoDB instance.

Image handling

When an admin creates a product, image upload and viewing need to be handled, being displayed as both a thumbnail and a main image on the product page. As we are using only MongoDB for storage, as well as accepting mainly JSON bodies across our APIs, there was difficulty storing binary data in our system. One option is to use Mongoose or GridFS (GridFS, 2021), which allows for storing larger files, in the case of GridFS, when storing over 16 MB. However, our database instance only allows for 500MB of storage, where storing high-definition large images is not necessary for our use case. To reduce the complexity of our project, where files would need to be stored as chunks and retrieved iteratively, we decided to encode our binary data as base64 before sending it to our backend. Base64 encoding and decoding is quite efficient, according to our tests,

as well as only promoting a 33% size increase overhead versus storing as a raw binary, which does not affect our system (Base64, 2021). This also means we can maintain a consistent JSON format in API calls, especially when editing and creating a product, where we can create more maintainable and portable frontend and backend code which we believe outweighs the overhead disadvantages. Further, MongoDB allows for specifying the fields we wish to query for, where we can exclude the image field when querying for products to increase efficiency. Using a separate database for image hosting such as Google Firebase could solve this issue, however the increase in complexity and a more than likely increased latency when querying two databases versus only our MongoDB instance deemed it unnecessary (Firebase, 2021).

Sharing API Wrapper instances

When using API wrappers such as Pymongo and Spotify, it is essential that a new wrapper instance isn't produced at each request. If multiple users use the site, this results in scaling inefficiency as a new connection is established on each instance creation. This problem is stated by pymongo: "Create this client **once** for each process, and reuse it for all operations. It is a common mistake to create a new client for each request, which is very inefficient.", where all wrappers used support connection pooling on the single instance (Pymongo, 2021). To combat this, we use the Borg pattern, which is a Python specific pattern, similar to the Singleton pattern, setting the internal dictionary to a saved state every time a new object is initialised, forcing the same API wrapper instances to be reused (Borg Pattern, 2002). From here, the internal client is abstracted from the backend developer such that the wrappers can be called without worrying about the specific internal implementation of each API e.g. db = MongoWrapper().client['vinyl_store'] for pymongo and sp = SpotifyWrapper().client for Spotify.

User Documentation/Manual

How to build, setup, configure, and use the system and functionalities

Configuration

Configuration options for this system can be found in the “credentials.json” file located in the “flask-app/credentials/” directory. Configuration options are as follows:

- “username” is the username of the MongoDB database account
- “password” is the password of the MongoDB database account
- “connection_string” is the MongoDB URI connection string used to connect to the database
- “discogs_user_token” is the Discogs API token used to pull mock data for the database from Discogs
- “spotify_client_id” is the application client ID required to use the Spotify Web API
- “spotify_client_secret” is the key used for secure API calls to Spotify
- “spotify_client_redirect” is address whitelisted with Spotify so that the app can be relaunched when the user logs in
- “spotify_client_cache” is the location of the cached Spotify oauth token
- “stripe_api_publishable_key” is the public Stripe API key used as an identifier for the Stripe account used
- “stripe_api_secret_key” is the secret Stripe API key used for API requests
- “mail_server” is the SMTP server used to send emails to users
- “mail_port” is the port of the SMTP server
- “mail_username” is the username (email address) of the email account used to send emails
- “mail_password” is the password of the email account

Setup

You can run the included script with “./build.sh” to build and run the system on a Linux computer or server. The system requires Python 3.8, virtualenv 20.x, and Node.js 12.x or higher to run.

Open “localhost:3000” in your browser to view the application.

Alternatively, you can manually install the dependencies and run the system by following the instructions below.

Backend Setup

The backend application requires Python 3.8 or higher to run.

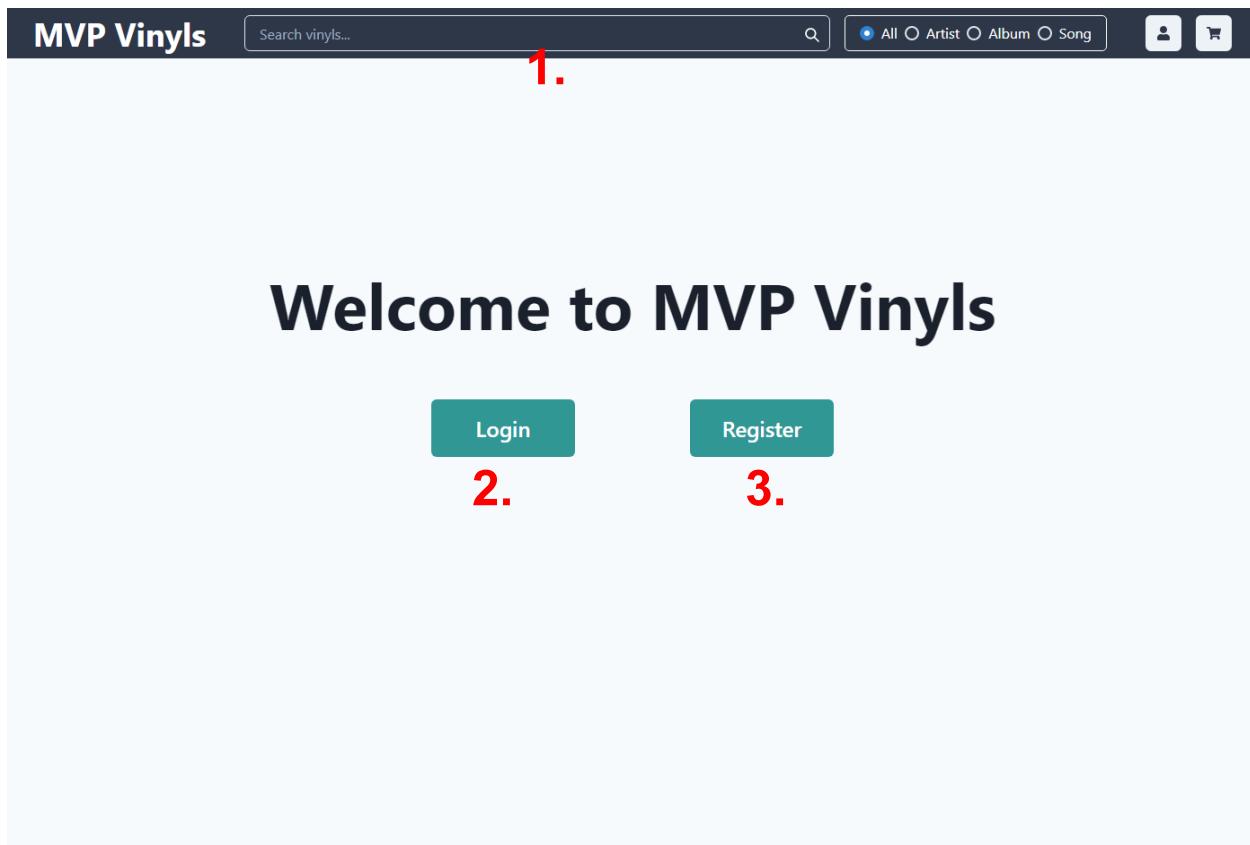
1. Install virtualenv. We recommend the latest stable version 20.4.4
2. Change directory into “flask-app”
3. Create a new virtual environment using command “python3 -m virtualenv venv”
4. Activate the virtual environment using “source venv/bin/activate”
5. Install the dependencies using “pip3 install -r requirements.txt”
6. Start the backend app using “python3 server.py”

Frontend Setup

The frontend application requires Node.js 12.x or higher to run.

1. Install Yarn package manager. We recommend the latest stable version v1.22.5
2. Change directory into “react-app”
3. Install the dependencies of the project using command “yarn” or “yarn install”
4. Start the frontend app by using command “yarn start”
5. View frontend app in browser at “localhost:3000”

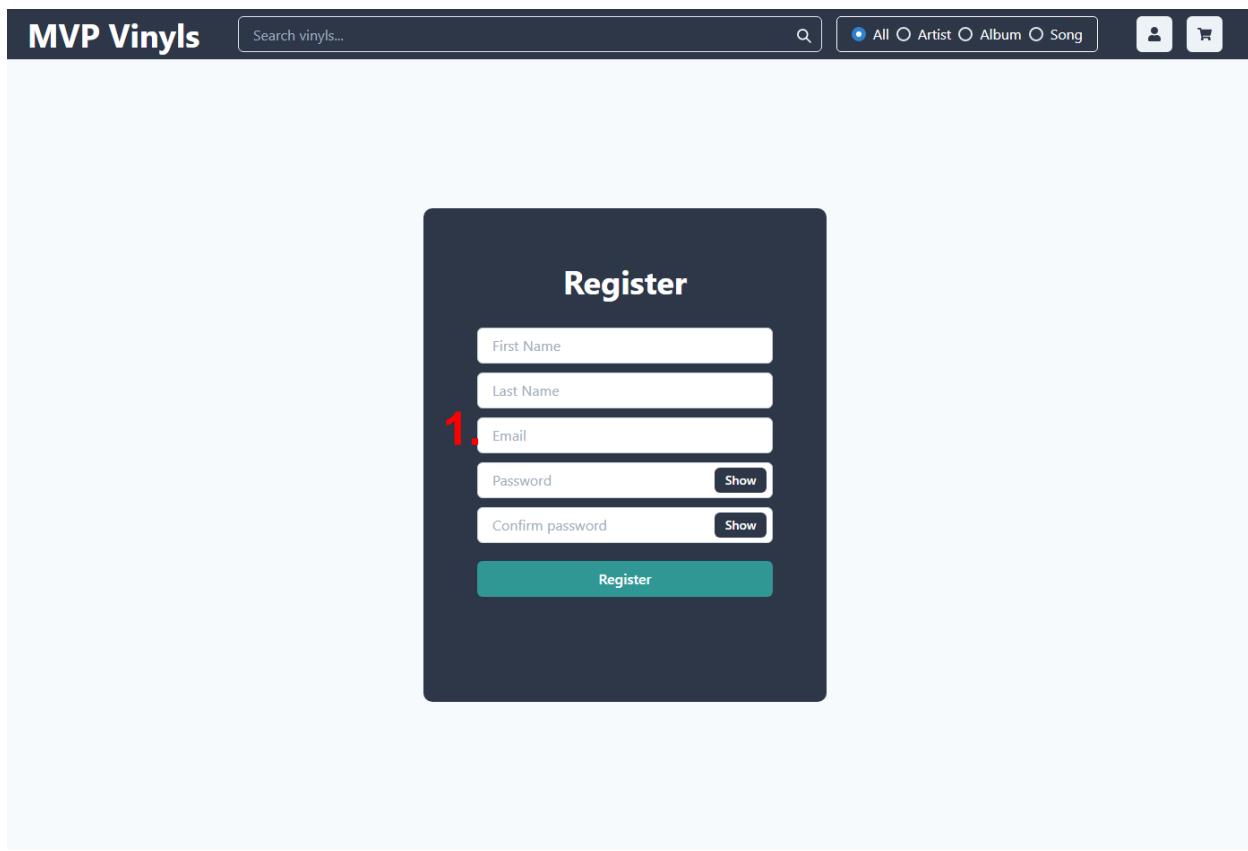
Usage and Functionalities



Landing Page

The MVP Vinyls landing page is the first page a user will see when using our website. Access to the rest of the website requires a user to be logged in. A user is initially not logged in.

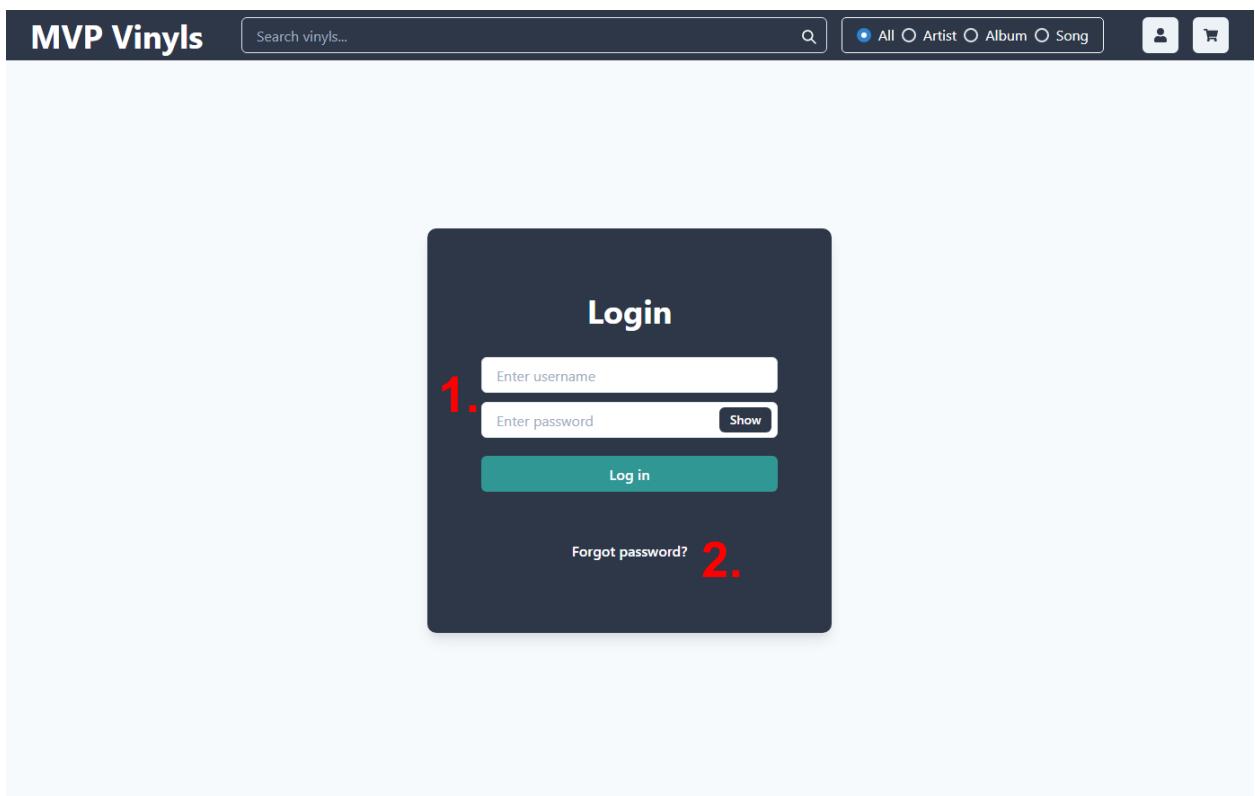
1. Interacting with the logo, search bar and cart in the navbar will display an alert informing the user they are not logged in and will present them with buttons linking to the login and register pages.
 - a. The profile button will display a dropdown with links to the login and register pages
2. Clicking the login button will redirect the user to the login page
3. Clicking the register button will redirect the user to the register page



Register Page

1. A user can create an account by entering their details in the register form. The inputs are first name, last name, email, password and confirm password.
 - a. Email must be unique and not already associated with an existing account
 - b. Password and confirm password must be the same

Clicking the register button will submit the form and the user will be redirected to the homepage on successful register. On an unsuccessful register an appropriate error message will be shown.



Login Page

1. A user can login to their account by entering their username - which is their email, and their password. Clicking the login button will redirect the user to the homepage on successful login. On an unsuccessful login an appropriate error message is shown.
2. In the case that a user forgets their password, the user can click the forgot password link to be redirected to the forgot password page where they will be able to reset their password

MVP Vinyls

Search vinyls...

All Artist Album Song

Forgot Password

Enter your email and we will send a code for you to reset your password.

1.

MVP Vinyls

Search vinyls...

All Artist Album Song

Reset Password

Enter Reset Code

2.

Enter New Password

Forgot/Reset Password Pages

1. A user can enter the email associated with their account to request a reset code for their password. Clicking continue submits the form. An email will be sent if the email is associated with a user. The page will automatically redirect to the reset password page on success. If the email does not exist in the database an appropriate error message is shown.
2. The user can enter the reset code they received in their email as well as the password they would like to change to. Clicking the reset password button will reset the account's password and log in the user on successful reset, redirecting them to the homepage. If the reset code is incorrect, the user will be shown an appropriate error message.

The screenshot shows the MVP Vinyls website homepage. At the top, there is a navigation bar with the site name "MVP Vinyls", a search bar, and filter options for "All", "Artist", "Album", and "Song". On the right side of the header, there are icons for user profile and shopping cart, with the number "5." indicating five items in the cart.

Below the header, a welcome message "Welcome to MVP Vinyls, Angus!" is displayed, followed by a call-to-action button "Continue with Spotify".

A section titled "Based off your past purchases" shows a message "No past orders! Shop suggestions below for recommendations." with a "Continue with Spotify" button.

A "Suggestions" section displays a grid of vinyl records:

- ASD3 ASDF \$10.00
- Another Face David Bowie \$33.00
- I Might Be Wrong - Live Record... Radiohead \$10.00
- ILFORD NICE GUM The Art Ensemble Of Chic \$92.00
- PERRY COMO SINGS Conversation with MVP Vinyls Chatbot
- John Fogerty John Fogerty \$68.00
- Risotto Fluke \$71.00
- Welcome To Wherever You Are INXS \$80.00
- Strictly Business EPMD \$84.00

A chatbot window is open on the right side of the page, showing a conversation with the MVP Vinyls Chatbot. The bot says: "Hi I'm the MVP Vinyls Chatbot! How can I help you today?". Below the bot, there is a message input field with placeholder text "To select a help option please type in the chat:" and a "Product, Delivery, Warranty" button. The message input field also has a placeholder "Write your message here".

Homepage

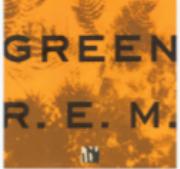
1. Upon logging into MVP Vinyls, the user is greeted with the homepage. By clicking on the 'Continue with Spotify' button, users are able to connect their Spotify account to MVP Vinyls to get recommendations based off of their top listened to genres.

-
2. Users are given recommendations based on their past orders, but in the event that a user has not made any previous orders, the site prompts users to shop the suggestions below.
3. Suggestions are given to the user for a convenient user browsing experience.
4. Clicking the chat icon in the bottom right opens the chatbot. The chatbot has three main help options: product, delivery and warranty. To begin interacting with the chatbot a user must first select a help option by typing into the chat one of the options. Once an option has been selected, the user can change options by typing one of the other options.
 - a. Product
 - i. A user can enter a query related to the product they are looking for. The chatbot will return the top three products related to your query. Clicking on a product will link you to the product page of the product.
 - b. Warranty
 - i. A user can enter the name of a product they want to know the warranty details for. Entering a product that they have ordered will calculate the warranty based on purchase date. A link is provided to the corresponding order. Entering a product they have not purchased will show the warranty of that product. A link is provided to the product page.
 - c. Delivery
 - i. A user can enter the order ID of one of their orders to show the expected delivery date of their order. A link is also provided that redirects the user to the order details page.
5. To log out, a user can click the profile button to be presented with a log out option. Clicking log out will redirect the user back to the landing page.

MVP Vinyls All | Artist | Album | Song  

Recommended for you, Angus!

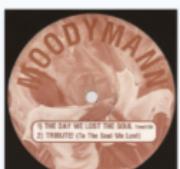
1. Based off your past purchases

 Another Face David Bowie \$33.00	 Green R.E.M. \$79.00	 I Might Be Wrong - Live Recor... Radiohead \$10.00	 Wild Horses (Acoustic Version)... The Rolling Stones \$50.00	 No Line On The Horizon U2 \$76.00
 Status Quo Status Quo \$19.00	 Alice Cooper's Greatest Hits Alice Cooper \$20.00	 Move Of Ten Autrechre \$2.00	 Coma / Spraycan Bad Company \$31.00	 Anthem Of The Sun The Grateful Dead \$5.00

2. Based off your Top Spotify Genres

 TRG Remixes TRG \$97.00	 Rock The House DJ Jazzy Jeff & The Fresh Prince \$23.00	 Stone Rollin' Raphael Saadiq \$96.00	 Strictly Business EPMD \$84.00	 These Are Special Times Céline Dion \$52.00
 Alme Makam \$33.00	 Do You Want The Truth Or So... Paloma Faith \$42.00			

Suggestions

 The Day We Lost The Soul Moodymann \$64.00	 Welcome To Wherever You Are INXS \$80.00	 Fat Larry's Skank - Remix Kodak Black & The Cubans \$3.00
---	---	--

Recommendations

- Once a user has made an order, recommendations are displayed on the homepage based on their past orders.

2. Once users are logged into Spotify, vinyl recommendations are displayed on the homepage. Giving recommendations tailored to their personal preferences makes the whole user experience more personalised and unique.

The screenshot shows the MVP Vinyls application interface. At the top, there is a navigation bar with the title "MVP Vinyls", a search bar containing "Search vinyls...", and filter options for "All", "Artist", "Album", and "Song". On the right side of the header are icons for user profile and shopping cart. A red number "1." is placed above the user profile section. The main content area is a white box titled "User Profile". Inside, there are three sections: "Name" (Angus Feng), "Email Address" (angus23@gmail.com), and "Shipping Address" (16 Test Street, Test city 2222, Test state, Test country). To the left of the profile box, a red number "2." is above the "User Settings" tab, and a red number "3." is to the right of the "Shipping Address" section. The left sidebar contains "User Management" with "Profile" and "Order history" links, and "User Settings" with "Edit profile", "Edit shipping address", and "Change password" links.

User Pages

1. A logged in user can view their user management and user settings pages by clicking the profile button. This will present a drop down with an option to view profile. Clicking the link will redirect the user to their user profile page
2. A user can visit different user management and settings pages by clicking on the links in the side tab. The side tab will remain but the page content will change.

User Profile Page

3. A user can view their name, email address and shipping address. If a shipping address has not been specified (such as on first login) then the text will display "None".

The screenshot shows the MVP Vinyls website interface. At the top, there is a navigation bar with the logo "MVP Vinyls", a search bar containing "Search vinyls...", and a dropdown menu with options "All", "Artist", "Album", and "Song". To the right of the search bar are icons for user profile and shopping cart. Below the navigation bar, on the left, is a sidebar titled "User Management" with links for "Profile" and "Order history". On the right, a modal window titled "Edit Profile" is open, showing fields for "First Name" (Angus), "Last Name" (Feng), and "Email Address" (angus23@gmail.com). A red number "1." is overlaid on the "Edit Profile" title. At the bottom of the modal is a "Save Changes" button.

Edit Profile Page

Clicking edit profile in the side tab will redirect the user to the edit profile page.

1. The form will be prefilled with the user's first name, last name and email address. The user can edit their details in the form. Clicking the save changes button will submit the form.

On a successful edit the user will be given a success alert. If an error occurs such as an empty field, invalid email address, existing email address etc, the user will be given an appropriate error message.

The screenshot shows the MVP Vinyls website interface. At the top, there is a navigation bar with the logo "MVP Vinyls", a search bar containing "Search vinyls...", and a dropdown menu with options "All", "Artist", "Album", and "Song". To the right of the search bar are icons for user profile and shopping cart.

The main content area has two side tabs: "User Management" (selected) and "User Settings". Under "User Management", there are links for "Profile" and "Order history". Under "User Settings", there are links for "Edit profile", "Edit shipping address", and "Change password".

The central area is titled "Order History". It displays a table with three rows of past orders:

ORDER #	DATE OF ORDER	ITEMS	ACTIONS
1. 607f8e91fee05df0cb121faf	04/21/2021		View Details
607fe1633f96cc648e1e7420	04/21/2021		View Details
607fe2f03f96cc648e1e74b5	04/21/2021		View Details

Order History Page

Clicking the order history tab on the side tab will redirect the user to the order history page.

1. The order history page displays all the user's past orders. It shows each order's order number, date of order, the items purchased in that order and the option to view further details about each order.

The screenshot shows the MVP Vinyls website's user management section. On the left, there's a sidebar with 'User Management' (Profile, Order history) and 'User Settings' (Edit profile, Edit shipping address, Change password). The main content area is titled 'Shipping Address'. It contains five input fields: 'Street Address' (16 Test Street), 'City or Town' (Sydney, marked with a red '1.'), 'Postcode' (2222), 'State' (NSW), and 'Country' (Australia). At the bottom is a teal 'Save Changes' button.

Edit Shipping Address Page

Clicking edit shipping address in the side tab will redirect the user to the edit shipping address page.

1. If a shipping address has not been previously supplied the form will initially have empty inputs. Otherwise, if previously supplied, the form will come prefilled with the user's street address, city, postcode, state and country. Clicking the save changes button will submit the form.

On a successful edit the user will be given a success alert. If an error occurs the user will be given an appropriate error message.

The screenshot shows the MVP Vinyls application's user management section. On the left, there's a sidebar with 'User Management' and 'User Settings' sections. Under 'User Management', there are links for 'Profile' and 'Order history'. Under 'User Settings', there are links for 'Edit profile', 'Edit shipping address', and 'Change password'. The main content area is titled 'Change Password' and contains two input fields: 'Current Password' and 'New Password', each with a 'Show' button to the right. Below these fields is a green 'Change password' button. A large red number '1.' is overlaid on the top-left portion of the form.

Change Password Page

Clicking change password in the side tab will redirect the user to the change password page.

1. The form does not come prefilled. A user can enter their current password as well as the new password they would like to change to. Clicking the show button next to the inputs will show both inputs. Clicking the change password button will submit the form.

On a successful edit the user will be given a success alert. If an error occurs the user will be given an appropriate error message.

MVP Vinyls

Search vinyls...

All Artist Album Song

Top Genres

- rock
- classic rock
- alternative rock
- album rock
- alternative metal

Sort Alphabetical

Exploring

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11.

1. Outside (The Nathan Adler D...
David Bowie
\$25.00

4AD Session
Daughter (2)
\$36.00

6 Months To X-Tinction
Evil Activities
\$47.00

A Night At The Playboy Mansion
Dimitri From Paris
\$37.00

Africafunk: The Original Sound...
Various
\$12.00

Aimé
Makam
\$33.00

Alice Cooper's Greatest Hits
Alice Cooper
\$20.00

All She Wants Is
Duran Duran
\$35.00

Another Face
David Bowie
\$33.00

Anthem Of The Sun
The Grateful Dead
\$5.00

ASD3
ASDF
\$10.00

Capture In Distress
Catscan
\$2.00

Showing 12 of 85 products

Previous Next Page

Search Page

1. Doing a search in the search bar will redirect the user to the search page. A user can complete a search by pressing enter or clicking the search button.
2. A user can specify a category within which they would like to perform a search for. The default is search by all. A user can select to search by all, artist, album or song. Searching in the navbar will then only search by that category
3. A user is presented with a list of products that match the search query. Doing an empty search will show a list of all products. A product has an album image, album name, artist and price.
4. A user can choose to sort their products alphabetically or by price. Selecting a sort option will update the products. The default option is alphabetical.

-
- 5. By default 12 products are shown per page. If all products can't fit on the page, the user can click the next page button to view the next page of products. Similarly they can click the previous button to go back to the previous page of products.
 - 6. A user can filter the products within the query they searched for by the top 5 genres within those products. Clicking a genre will add the genre to the filter list and update the products. A user can filter by multiple genres to specify their search. Multiple genres will filter by logical AND.

Clicking a genre already in the filter list will remove the genre from the filter list and update the products.

MVP Vinyls Search vinyl... All Artist Album Song  

Home / Duran Duran / All She Wants Is

Duran Duran \$35.00
All She Wants Is

In Stock  1.

Description **2.** Specification

Tracklist

- All She Wants Is (45 Mix)
- I Believe / All I Need To Know (Medley)

Similar Products **3.**

| | | | | |
|-------------------------------------|-------------------------|--------------------------------|-------------------------------------|------------------------------|
| Welcome To Wherever You Are
INXS | Different Light Bangles | Coma / Spraycan
Bad Company | Scarecrow
John Cougar Mellencamp | John Fogerty
John Fogerty |
|-------------------------------------|-------------------------|--------------------------------|-------------------------------------|------------------------------|

Individual Vinyl Page

Clicking on a vinyl will bring the user to that vinyl's individual page which displays the vinyl's album name, artist's name and price.

- When a product is in stock, users are able to add the product into their cart by clicking on the 'Add to cart' button. Users are able to specify the amount of that vinyl they want by either typing a number or incrementally increasing or decreasing the amount by clicking on the input stepper button to the right of the 'Add to cart' button.

Each time a product is added to the cart, the system will give the user feedback indicating as such.

- Users can switch between the 'Description' or 'Specification' tab to view detailed information about the vinyl. The description tab presents a user with the tracklist

of that specific vinyl and switching to the specification tab will provide information about the vinyl's SKU number and what genres the vinyl falls under.

- Products similar to the vinyl are displayed at the bottom, giving users the option to discover other products in the database that are aligned with their tastes.

MVP Vinyls Search vinyls... All Artist Album Song 1.

Shopping Cart

| Product | Description | Quantity | Subtotal |
|---------|---|----------|----------|
| | 4AD Session
Daughter (2) | 3 | \$108.00 |
| | 6 Months To X-Tinction
Evil Activities | 1 | \$47.00 |

Order Summary

| Item | Quantity | Subtotal |
|---------------------------|-----------------|----------|
| 1. 4AD Session | 3 | \$108.00 |
| 2. 6 Months To X-Tinction | 1 | \$47.00 |
| Total Items: | 4PCS | |
| Grand Total: | \$155.00 | |

Buttons:

- Update 2. (next to the first item)
- Delete 3. (next to the first item)
- Checkout 4. (green button)
- Continue Shopping 5. (green button)

Shopping Cart Page

- Users are able to view their cart by clicking on the shopping icon in the top right corner of the page.
- On the shopping cart page, the vinyls that were added into the cart by the user are displayed in chronological order. From here, users are able to edit their cart by either typing the new amount or incrementally increasing or decreasing the amount by clicking on the input stepper. Clicking on the 'Update' button will update the cart.
- Users are also able to delete a product completely from their cart by clicking on the 'Delete' button in the top right corner of the vinyl card. When an item is deleted, the cart updates dynamically and this change is reflected when looking at the order summary.
- Users can proceed to purchase the products in their cart by clicking on the 'Checkout' button which will lead them to the payment process.
- Users are also able to continue browsing products by clicking on the 'Continue Shopping' page which will lead them to the homepage.

MVP Vinyls Search vinyl... All Artist Album Song

Checkout

1. Delivery Options 2.

First Name *

Last Name *

Street Address * 1.

City *

Postcode *

State *

Email *

3.

Checkout Delivery Options Page

1. A user can enter the delivery details for their order. The form inputs are required.
2. For convenience, a user can click the use saved button to enter the details associated with their account if those are the details they wish to use for their order.
3. Clicking the continue button submits the delivery options for the order and redirects to the order summary page

2. Order Summary

Cart

| | | |
|--|--|--|
| | 4AD Session
Daughter (2) | \$36.00
Quantity: 3
Subtotal: \$108.00 |
| | 6 Months To X-Tinction
Evil Activities | \$47.00
Quantity: 1
Subtotal: \$47.00 |

Shipping details

16 Test Street, Sydney, NSW 2222
angus23@gmail.com

Order Total: \$155.00

Go back **2.** **Proceed to payment**

Checkout Order Summary Page

1. The order summary displays a summary of a user's order. It displays each product in the cart including the album name, artist name, price, quantity and subtotal.
It displays the shipping details entered in the delivery options page
It displays the total price of the order
2. Clicking proceed to payment redirects the user to the Stripe payment page

← MVPs Vinyl TEST MODE

1. Pay MVPs Vinyl **US\$155.00**

| | |
|---------------------------------|------------------------------|
| 4AD Session
Qty 3 | US\$108.00
US\$36.00 each |
| 6 Months To X-Tinction
Qty 1 | US\$47.00 |

Powered by stripe | [Terms](#) [Privacy](#)

2. Pay with card

Email

Card information

| | |
|---------|-----|
| MM / YY | CVC |
|---------|-----|

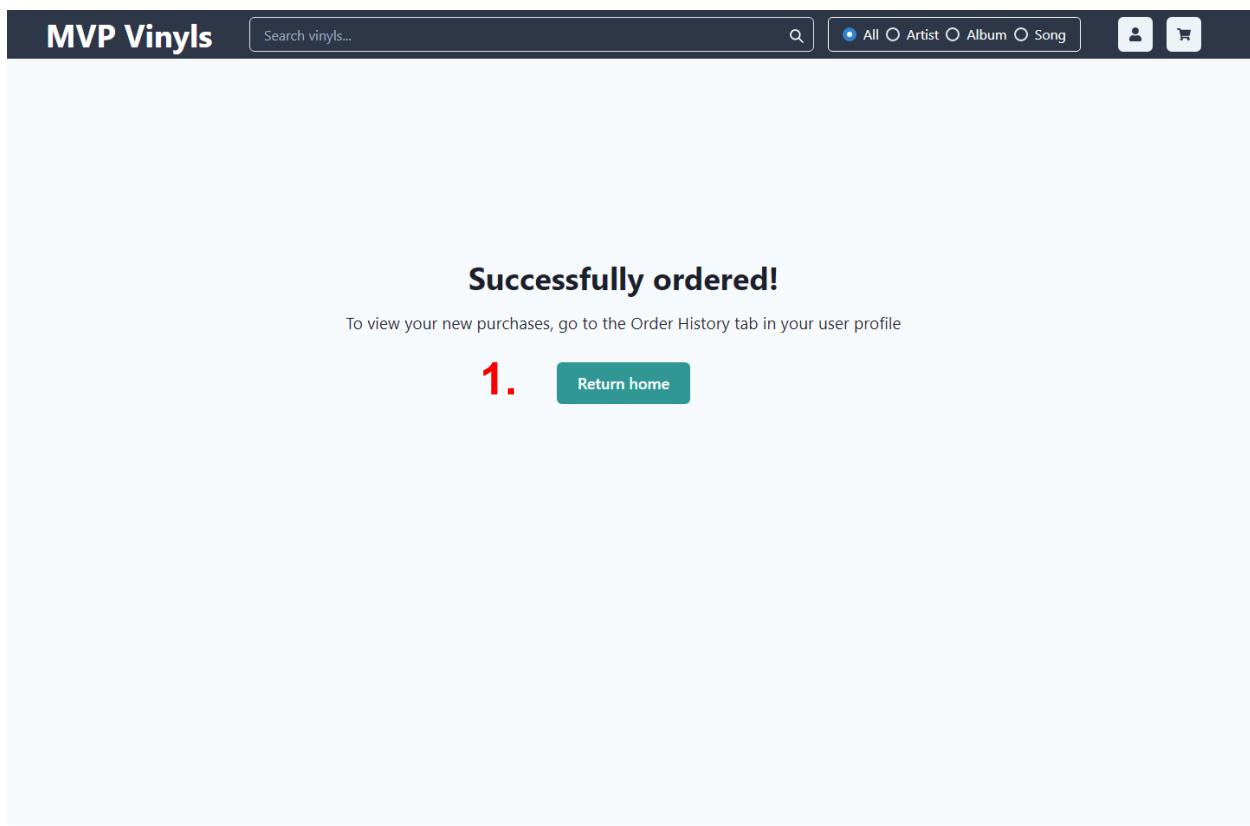
Name on card

Country or region Australia

Pay US\$155.00

Stripe Payment Page

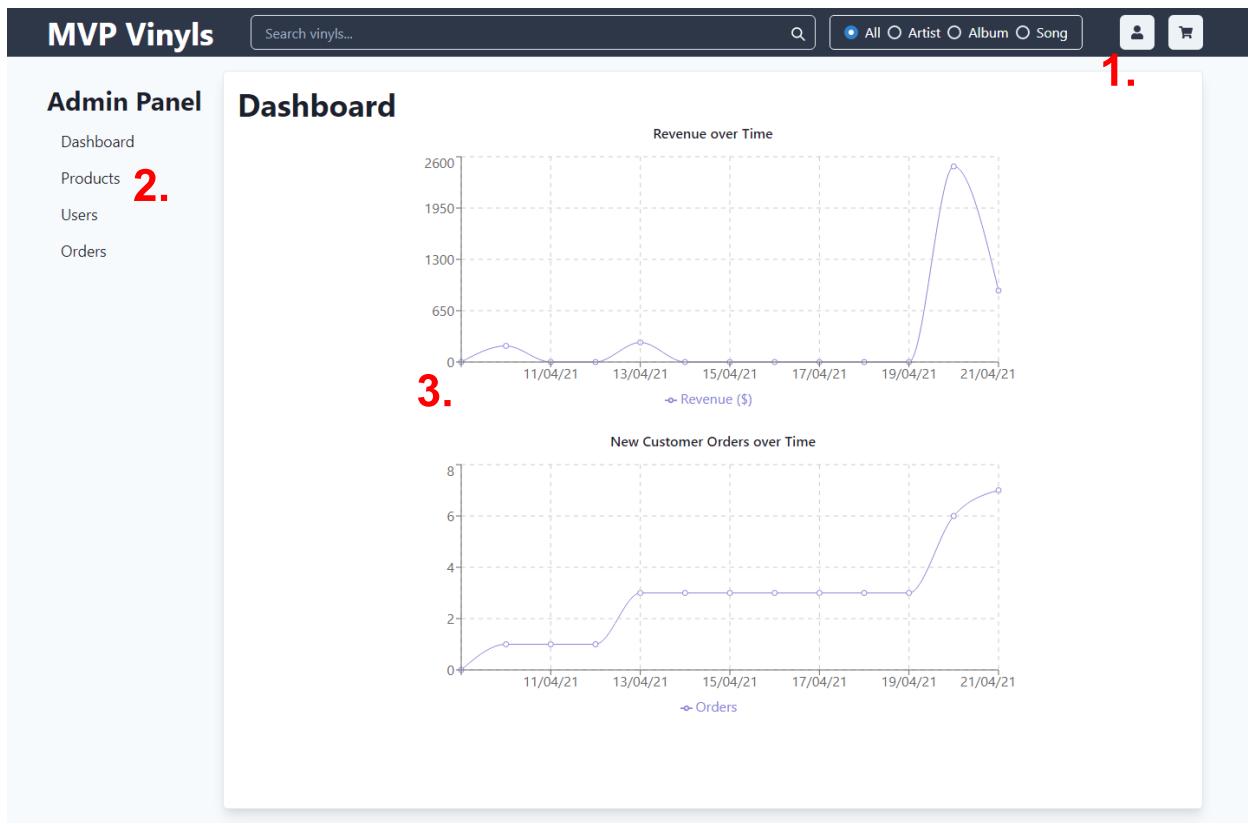
1. The total price of the order is displayed. The items to be ordered are also displayed showing the album name, quantity, subtotal and individual price if more than one is being ordered.
2. A user can enter their payment details to pay for their order. For the purpose of demoing a successful card number is 4242 4242 4242 4242. An unsuccessful card number is 4000 0000 0000 9995.



Order end page

From the stripe payment page, a successful payment will redirect to the order end page. If the order is unsuccessful an error message is shown instead.

1. Clicking the return home button will redirect the user back to the home page



Admin Panel

1. A logged in user that has admin privileges is able to view the admin panel by clicking on the user profile button in the top right corner of the page. This will display a drop down menu with the option to direct the admin to the admin panel.
2. An admin can access various admin functions by clicking through the tabs on the side menu to the left.

Admin Dashboard

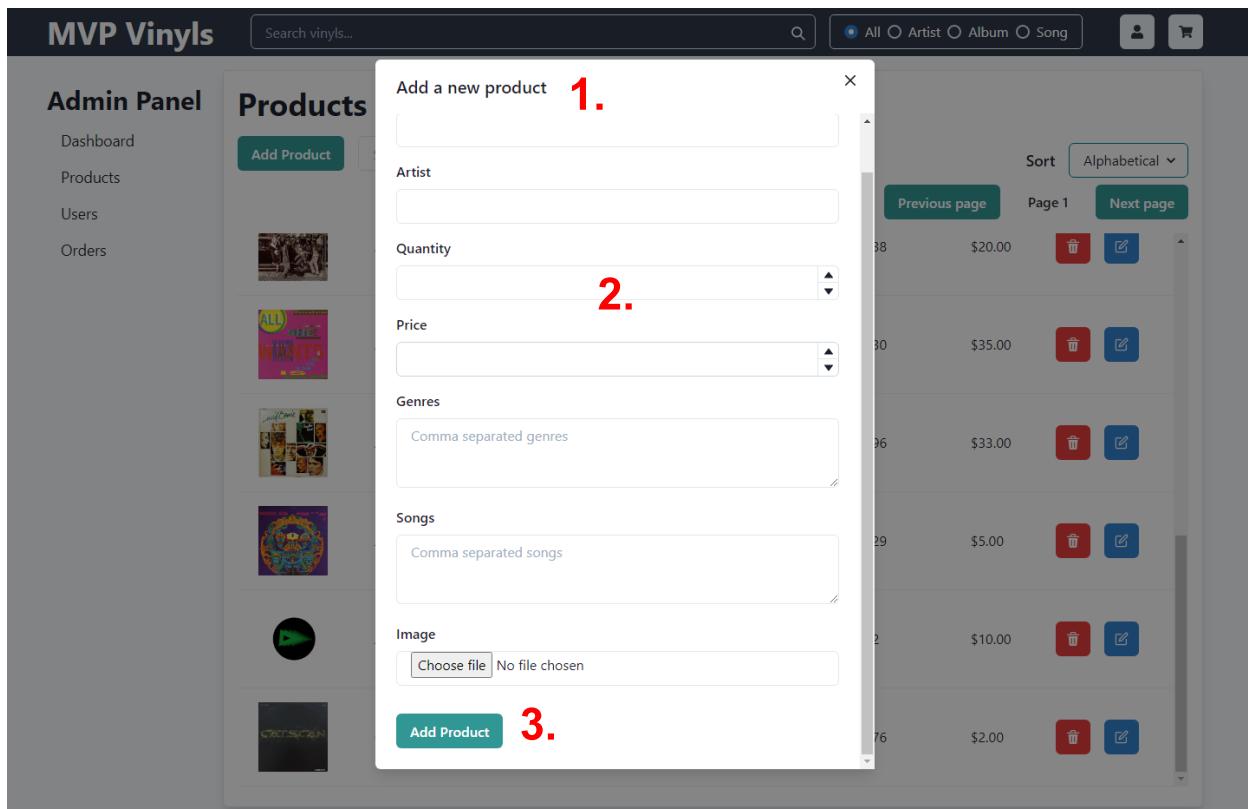
3. The admin dashboard provides admins with business insights through the form of graphs. The first graph illustrates revenue over time and the second graph represents new customer orders over time. Hovering over each graph will present exact figures to an admin.

| ALBUM COVER | ALBUM NAME | ARTIST | QUANTITY | PRICE (\$) | ACTIONS |
|-------------|--|--------------------|----------|------------|---------------------------------|
| | 1. Outside (The Nathan Adler Diaries: A Hyper Cycle) | David Bowie | 7 | \$25.00 | 5. 6. |
| | 4AD Session | Daughter (2) | 39 | \$36.00 | 5. 6. |
| | 6 Months To X-Tinction | Evil Activities | 47 | \$47.00 | 5. 6. |
| | A Night At The Playboy Mansion | Dimitri From Paris | 44 | \$37.00 | 5. 6. |
| | Africafunk: The Original Sound Of 1970s Funky Africa | Various | 62 | \$12.00 | 5. 6. |

Admin Products Page

An admin can view a list of all the products currently in the database from the admin products page. They can see a product's album cover, album name, artist, quantity and price.

1. Clicking on the add product button presents the admin with a modal form that allows them to add a product
2. An admin can search vinyls by typing in the search bar. The search automatically enters the search without the need to press the search button. However, the search button can still be pressed to complete the search if it is too slow.
3. An admin can sort the products by alphabetical order or price order
4. An admin can view different pages of products if they don't all fit on the page of 12 products.
5. Clicking the delete button prompts an alert dialog confirming if they want to delete the product. Clicking delete on the alert then deletes the product.
6. Clicking the edit modal button presents the admin with a modal form that allows them to edit the product



Add/Edit Product Modal

1. The add product modal is identical to the edit product modal. The difference is that the edit product modal comes prefilled with the product's details.
2. An admin can enter the details of the product. Genres and songs are entered as comma separated strings.
3. Clicking on the add product button will add a product to the database. If editing a product, clicking on the edit product button will edit the product's details.

On successful add/edit the modal is closed and a success alert is displayed on the admin products page. If unsuccessful an appropriate error alert is displayed.

Admin Panel

Users

| FIRST NAME | LAST NAME | USER EMAIL | ROLE | ACTIONS |
|------------|-----------|---------------------------|-------|----------------|
| admin | admin | benadmin@gmail.com | Admin | Demote |
| Alex | Test | bekir41176@whipjoy.com | User | Promote |
| angela | yao | newemail13@test.com | User | Promote |
| Angela | Yao | newemail10@test.com | User | Promote |
| Angela | Yao | angela@test.com | Admin | Demote |
| Angela | Yao | angela2@test.com | Admin | Demote |
| Angela | Yao | newemailtest@test.com | User | Promote |
| Angela | Yao | angela3@test.com | Admin | Demote |
| Angela | Yao | newcustomertest@gmail.com | User | Promote |

Admin Users Page

An admin can view a list of all users currently in the database. An admin can see a user's first name, last name, user email and role.

1. An admin can search users by typing in the search bar. The search automatically enters the search without the need to press the search button. However, the search button can still be pressed to complete the search if it is too slow.
2. An admin can view different pages of users if they don't all fit on the page of 10 users
3. Clicking the promote/demote button prompts an alert dialog confirming if they want to promote/demote the user. Clicking promote/demote on the alert then promotes/demotes that user. If successful a success alert is shown. The user's role and button is updated accordingly in the table.

| ORDER # | SHIPPING EMAIL | DATE OF ORDER | ACTIONS | FULFILLMENT STATUS |
|--------------------------|-------------------|----------------------|--|--------------------|
| 607fe2f03f96cc648e1e74b5 | angus23@gmail.com | 04/21/2021, 18:31:44 | <button>View Order</button> <button>Fulfill Order</button> | |
| 607fe1633f96cc648e1e7420 | angus23@gmail.com | 04/21/2021, 18:25:07 | <button>View Order</button> <button>Fulfill Order</button> | |
| 607f8e91fee05df0cb121faf | angus23@gmail.com | 04/21/2021, 12:31:45 | <button>View Order</button> | Fulfilled |
| 607e626b5ef8d490077cd57a | userboi@gmail.com | 04/20/2021, 15:11:07 | <button>View Order</button> | Fulfilled |
| 607e5147951f23f6e155bf91 | email | 04/20/2021, 13:57:59 | <button>View Order</button> <button>Fulfill Order</button> | |
| 607e511d8ed03c8fdfcc607 | email | 04/20/2021, 13:57:17 | <button>View Order</button> <button>Fulfill Order</button> | |
| 607e50ef28aba796fbf2bbfc | email | 04/20/2021, 13:56:31 | <button>View Order</button> <button>Fulfill Order</button> | |
| 607e50b1d684727a1d315c83 | email | 04/20/2021, 13:55:29 | <button>View Order</button> <button>Fulfill Order</button> | |
| 607e4fd7953b652737f3f5d2 | email | 04/20/2021, 13:51:51 | <button>View Order</button> <button>Fulfill Order</button> | |

Admin Orders Page

An admin can view a list of all orders currently in the database. An admin can see an order's order number, user's shipping email, date of order and its fulfilled status

1. An admin can search orders by typing in the search bar. The search automatically enters the search without the need to press the search button. However, the search button can still be pressed to complete the search if it is too slow.
2. An admin can view different pages of orders if they don't all fit on the page of 10 orders
3. Clicking the view order button redirects the admin to the order details page of that product
4. Clicking the fulfill order button updates the order's fulfilled status to be true. If successful a success alert is shown.

MVP Vinyls

Search vinyls...

All Artist Album Song

User Cart

Order Details

| | | | | | |
|--|--|--|----------------------|-----------------|-------------------|
| Order Number: | 607fe2f03f96cc648e1e74b5 | Order Placed: | 04/21/2021, 18:31:44 | Shipping Email: | angus23@gmail.com |
| Shipping Details: Angus Feng, 16 Test Street, Sydney 2222, NSW | | | | | |
| Fulfilled: | False 1. | | | | |
| Order total: | \$155.00 | | | | |
| Items: | | | | | |
| | 4AD Session
Daughter (2) | \$36.00
Quantity: 3
Subtotal: \$108.00 | | | |
| | 6 Months To X-Tinction
Evil Activities | \$47.00
Quantity: 1
Subtotal: \$47.00 | | | |

2. [Return to previous](#)

Order Details Page

1. A user can view an order's order number, order date and time, shipping email, shipping details, fulfilled status and a list of items in the order
2. Clicking the return to previous button returns the user to the previous page

References

Amazon Web Services (2021). Cloud Computing Services. Retrieved from
<https://aws.amazon.com/>

Base64 (2021). Represent binary data in an ASCII string format by translating it into a radix-64 representation. Retrieved from
<https://developer.mozilla.org/en-US/docs/Glossary/Base64>

Borg Pattern (2002). Avoiding the Singleton Design Pattern with the Borg Idiom. Retrieved from
<https://www.oreilly.com/library/view/python-cookbook/0596001673/ch05s23.html>

Chakra UI (2021). A simple, modular and accessible component library that gives you the building blocks you need to build your React applications. Retrieved from
<https://chakra-ui.com/>

Cosine Similarity (2012). The similarity between two vectors of an inner product space. Retrieved from
<https://www.sciencedirect.com/topics/computer-science/cosine-similarity>

Discogs (2021). David Bowie & Morrissey - Cosmic Dancer. Retrieved from
<https://www.discogs.com/David-Bowie-Morrissey-Cosmic-Dancer/release/17349874>

Discogs (2021). Discogs Help Center. Retrieved from
<https://support.discogs.com/hc/en-us>

Discogs (2020). Discogs Mid Year Report 2020. Retrieved from
<http://blog.discogs.com/wp-content/uploads/2020/08/discogs-mid-year-report-2020.pdf>

Discogs_client (2021). Discogs API client for Python. Retrieved from
https://github.com/joalla/discogs_client

DNSPython (2021). dnspython. Retrieved from
<https://www.dnspython.org/>

Firebase (2021). Firebase helps you build and run successful apps. Retrieved from
<https://firebase.google.com/>

Flask (2021). The Python micro framework for building web applications. Retrieved from
<https://flask.palletsprojects.com/en/1.1.x/>

Flask-Cors (2021). Flask-CORS. Retrieved from
<https://flask-cors.readthedocs.io/en/latest/>

Flask-Mail (2021). Flask-Mail. Retrieved from
<https://pythonhosted.org/Flask-Mail/>

Forbes (2018). Vinyl is Bigger Than We Thought, Much Bigger. Retrieved from
<https://www.forbes.com/sites/billrosenblatt/2018/09/18/vinyl-is-bigger-than-we-thought-much-bigger/?sh=1cd213041c9c>

Google Cloud Platform (2021). Cloud Computing Services. Retrieved from
<https://cloud.google.com/>

GridFS (2021). Store large binary objects (e.g. files) in MongoDB. Retrieved from
<https://pymongo.readthedocs.io/en/stable/examples/gridfs.html>

Jellyfish (2021). jellyfish 0.8.2. Retrieved from
<https://pypi.org/project/stripe/>

Joalla (2021). python3-discogs-client. Retrieved from
https://github.com/joalla/discogs_client

MongoDB (2021). The most popular database for modern apps. Retrieved from
<https://www.mongodb.com/>

Numpy (2021). The fundamental package for scientific computing with Python. Retrieved from
<https://numpy.org/>

OAuth 2.0 (2021). The industry-standard protocol for authorization. Retrieved from
<https://oauth.net/2/>

Pandas (2021). A fast, powerful, flexible and easy to use open source data analysis and manipulation tool. Retrieved from
<https://pandas.pydata.org/>

Paypal (2021). PayPal Merchant Fees. Retrieved from
<https://www.paypal.com/au/webapps/mpp/merchant-fees>

Prettier (2021). What is Prettier? Retrieved from
<https://prettier.io/docs/en/index.html>

PyJWT (2021). Welcome to PyJWT. Retrieved from

<https://pyjwt.readthedocs.io/en/stable/>

Pymongo (2021). Python driver for MongoDB. Retrieved from
<https://pypi.org/project/pymongo/>

Reactjs (2021). A JavaScript library for building user interfaces. Retrieved from
<https://reactjs.org/>

Reactjs (2021). Create a New React App. Retrieved from
<https://reactjs.org/docs/create-a-new-react-app.html>

React-Icons (2021). React Icons. Retrieved from
<https://react-icons.github.io/react-icons/>

React Chatbot Kit (2021). React-chatbot-kit. Retrieved from
<https://fredrikoseberg.github.io/react-chatbot-kit-docs/>

React Router (2021). React Training: React Router. Retrieved from
<https://reactrouter.com/>

Recharts (2021). Recharts: A composable charting library built on React components. Retrieved from
<https://recharts.org/en-US/>

Scikit-learn (2021). Machine learning in Python. Retrieved from
<https://scikit-learn.org/>

Styled Components (2021). Visual primitives for the component age. Retrieved from
<https://styled-components.com/>

Spotify (2021). A lightweight Python library for the Spotify Web API. Retrieved from
<https://spotipy.readthedocs.io/en/latest/>

Square (2021). Payments Pricing with Square APIs and SDKs. Retrieved from
<https://developer.squareup.com/docs/payments-pricing>

Stripe (2021). Pricing. Retrieved from
<https://stripe.com/en-au/pricing>

Stripe (2021). React Stripe.js Reference. Retrieved from
<https://stripe.com/docs/stripe-js/react>

Stripe (2021). stripe 2.56.0. Retrieved from
<https://pypi.org/project/stripe/>

Stripe-python (2021). Stripe API documentation. Retrieved from
<https://stripe.com/docs/api>

TF-IDF (2019). A statistical measure that evaluates how relevant a word is to a document in a collection of documents. Retrieved from
<https://monkeylearn.com/blog/what-is-tf-idf>

The Recording Industry Association of America (RIAA) (2018). 2018 RIAA Music Industry Revenue Statistics. Retrieved from
<https://www.riaa.com/reports/2018-riaa-shipment-revenue-statistics-riaa/>

Virtualenv (2021). virtualenv. Retrieved from
<https://virtualenv.pypa.io/en/latest/>

Werkzeug (2021). Werkzeug. Retrieved from
<https://werkzeug.palletsprojects.com/en/1.0.x/>