

Lab 1

Lab 1 Draft - Update Section 1

Victoria Crouthamel

CS 411W

Thomas J. Kennedy

September 11, 2024

Version 1

Table of Contents

1. Introduction.....	2
2. Product Description.....	3
2.1 Key Product Features and Capabilities.....	3
2.2 Major Components.....	6
3. Identification of Case Study.....	8
4. Product Prototype Description.....	9
4.1. Prototype Architecture (Hardware/Software).....	10
4.2. Prototype Features and Capabilities.....	11
4.3. Prototype Development Challenges.....	12
5. Glossary.....	13
6. References.....	14

Listing of Figures

Figure 1: Competition Matrix.....	5
Figure 2: Major Functional Components Diagram.....	7
Figure 3: Prototype Comparison Table.....	9

1. Introduction

With today's society centered around consumerism, credit cards not only offer convenience but can also come with rewards and benefits tailored to get people to apply and encourage spending. According to a report by Experian in 2024, the average American holds approximately four credit cards, each with a different set of rewards and benefits. These options, while beneficial, introduce complex challenges for consumers: maximizing the benefits received from each card and program based on their specific purchases.

The scope of credit card and vendor rewards is diverse. Not only can card's rewards program vary, but each major industry also advertises their own. For instance, one card might offer cash back at grocery stores, while another may reward points when booking flights. This gets more complex when considering vendor-specific rewards, which typically aren't integrated into traditional credit card management tools. Current solutions in the market, although aimed at simplifying this decision-making process, often don't account for rewards not linked to a credit card. Many of these companies require users to share sensitive information, tend to operate with biases due to sponsored content, and can clutter the user experience with unnecessary features and ads.

Identifying these gaps, our team has developed "Rewards Maximizer," a web-based application designed to enable the management of credit card and vendor-specific non-credit card rewards. This solution offers the ability to optimize spending decisions, ensuring consumers can make the most out of their financial choices. Rewards Maximizer will offer the standard major credit card comparison features and built-in wallet but takes this a step forward by also providing information regarding vendor-specific rewards. This system is meant to show users how to get the most out of their money as they continue to shop at their favorite places.

2. Product Description

Rewards Maximizer is a web application that optimizes user purchases by comparing credit card rewards and vendor-specific deals/rewards. It aims to provide users with the best possible rewards for their purchases by utilizing the digital wallet system and recommendation algorithms. This will be accomplished by allowing the user to enter credit and rewards in their possession into a wallet inside the program. The customer will then enter what they are trying to buy, which will prompt Rewards Maximizer to search for potential purchases from different businesses and return the credit and rewards cards that will award the most points or discounts to the searched item.

2.1 Key Product Features and Capabilities

Figure 1 outlines all key features offered compared to competitors. The first features the application utilizes is crowdsourcing meant to enhance its database with real user experiences, helping users make informed decisions based on collective knowledge, and encouraging engaging with one another.

The application's search and comparison tool allows users to find and evaluate various credit cards and their associated rewards. This feature assists users in selecting the card that offers the most benefits for their specific purchases, addressing a common challenge among consumers who wish to maximize their rewards but may lack the time or resources to manually compare numerous offers. This also includes digital wallet functionality that centralizes all card and rewards information. This wallet stores current card/rewards data and integrates with the application's recommendation systems to suggest the most beneficial cards based on the user's

entered purchases. This feature simplifies the user's experience by providing a one-stop solution for managing multiple cards and rewards systems.

Notifications and alerts are another key feature of the application, keeping users up to date on new or changing rewards and deals, like other competitors. However, Rewards Maximizer enhances this feature by also including updates about the company and any changes in the terms or availability of specific rewards programs, ensuring users are always informed and able to make the best decisions regarding their rewards strategies.

Additionally, Rewards Maximizer is committed to user privacy and security by only using anonymous data for its analytical processes. This approach ensures that personal information is protected while still allowing the application to offer personalized recommendations based on aggregated user data.

[This space intentionally left blank]

Features	Rewards Maximizer	PayPal Honey	Rakuten	NerdWallet	Rocket Money
Crowdsourcing	✓				
Search and compare credit cards	✓			✓	
Wallet functionality	✓			✓	✓
Gives cash back on purchases		✓	✓		
Search web and applies coupons		✓			
Exclusive use of anonymous data for analysis	✓				
Integration of vendor-specific deals and reward systems	✓	Partial	Partial		
Company Updates	✓	Partial	Partial		
Search company rewards and loyalty programs	✓			Partial	
Notifications and Alerts	✓	✓	✓	✓	✓

Figure 1: Competition Matrix

2.2 Major Components

The architecture of "Rewards Maximizer" is designed to ensure performance and user-friendly interactions. The application is divided into three main components: Front-End, Back-End, and Testing, each fulfilling critical roles in the operation of the service.

Front-End: The user interface of Rewards Maximizer is built using HTML, CSS, and JavaScript, which are the core technologies used for web development. These tools provide the foundation for creating the user interface. Bootstrap is used as the CSS framework to ensure that the design is consistent across different devices and screen sizes. Additionally, Ember.js is utilized to develop the web application, offering a structured framework that supports more complex user interactions and data management on the client side.

Back-End: The server-side operations are completed using Rust. Rust is used for backend development and for integrating with the database that stores the web scraping data. The web framework Rocket, which is also built in Rust, manages web application processes and simplifies the configuration of routing and security settings. The application's data is stored and managed in SQLite, an open-source relational database management system that supports queries and data storage. Diesel, an ORM for Rust, is incorporated to facilitate communication between the application and the database, streamlining database interactions and data manipulation. The backend also utilizes Python along with BeautifulSoup, a Python library for parsing HTML and XML documents. This setup is used for the web crawler component of the application, which gathers and updates information on credit card rewards from various online sources.

Testing: QUnit is used for creating tests, specifically for unit testing the JavaScript code within the Ember.js framework. This testing framework helps in validating the functionality of

individual parts of the front-end application, verifying they perform as expected before the application is deployed. The integrated structure of these components is outlined in Figure 2, which provides a visual representation of how they interact within the application's architecture.

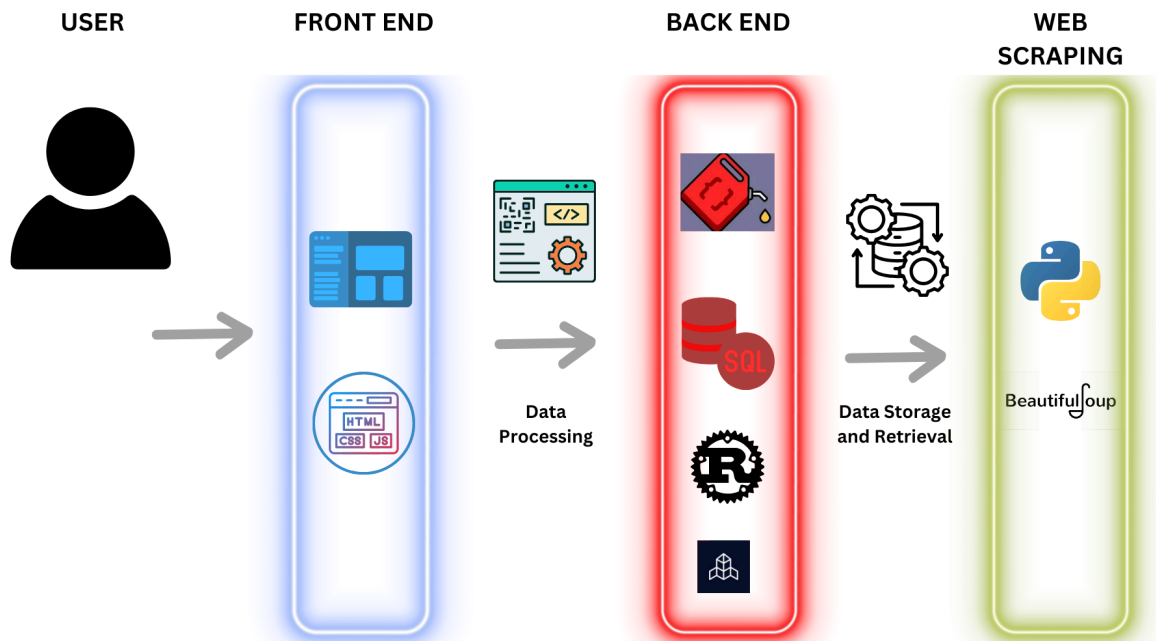


Figure 2: Major Functional Components Diagram

[This space intentionally left blank]

3. Identification of Case Study

- A. The application has the potential to be used by any consumer making purchases with a credit card or using vendor rewards programs.
- B. It is most likely to be used by the subset of these people who are money conscious and like to use and keep track of the kinds of cumulative small savings Rewards Maximizer can provide.
- C. It is possible the application may be used in the future by larger scale companies needing to buy products in bulk.

[This space intentionally left blank]

4. Product Prototype Description

- A. The prototype will implement:
1. Account management
 2. Login authentication
 3. The card and rewards search system
- B. The web crawler will be operating with an incomplete test dataset rather than live scraping multiple different financial sites.
- C. The card optimization algorithm will be simulated on test data rather than fully implemented with real up-to-date information.

Features	Real World Product	Prototype
User Account Setup	Fully Implemented	Not Implemented
Login Authentication	Fully Implemented	Not Implemented
Wallet Functionality	Fully Implemented	Not Implemented
Current Purchase Search	Fully Implemented	Not Implemented
Card Optimization Algorithm	Fully Implemented	Not Implemented
WebCrawler	Fully Implemented	Not Implemented
Feedback System	Fully Implemented	Not Implemented

Figure 3: Prototype Comparison Table

4.1. Prototype Architecture (Hardware/Software)

A. Hardware:

1. Basic server to host the prototype application.

B. Software

1. Front-End:

HTML, CSS, JavaScript, Bootstrap, Ember.js

2. Back-End:

Rust, Rocket, MySQL, Diesel

3. Web Scraper:

Python, BeautifulSoup

4. Testing:

JUnit

C. Prototype MFCD

The prototype will have a multi-functional component diagram (MFCD) similar to the final product but with some modules simplified or simulated:

1. User Interface (UI): Allows users to interact with the application, access the digital wallet, search products, and provide feedback.
2. Data Layer: Stores user information, card details, and interactions with the web scraper and card optimization algorithm.
3. Web Scraper: Operates with a test dataset instead of live data scraping.
4. Card Optimization Algorithm: Simulated using test data to demonstrate functionality.

4.2. Prototype Features and Capabilities

A. Key Features

1. User Account Setup and Login Authentication: Demonstrates basic user management, verifying users can create accounts and authenticate.
2. Card and Rewards Search System: Allows users to search for cards and rewards based on data.
3. Digital Wallet Functionality: Enable users to add and manage cards/rewards in a digital wallet.
4. Card Optimization Algorithm: Simulated on test data to show the ability to recommend the best card/rewards based on purchases.
5. Web Scraper: Runs on a test dataset to validate the concept of gathering data from various sources.

B. Risk Mitigation

1. Testing key features early in the development process allows for improvements to the overall design based on feedback and results.
2. Identifying technical challenges and the limitations of the stack that was chosen helps refine later.
3. Development and feedback will give the opportunity to implement features that bring in revenue.

4.3. Prototype Development Challenges

The development of the prototype is expected to encounter several challenges:

- A. Knowledge Gaps: Initial lack of knowledge on the intricacies of credit card rewards and vendor rewards programs may require additional research and learning.
- B. Technical Expertise: Inexperience with certain tools and technologies, particularly Rust, may pose development hurdles.
- C. Data Integration: Ensuring accurate simulation of the card/rewards optimization algorithm with test data instead of real-time information.

[This space intentionally left blank]

5. Glossary

Term	Definition
Application Programming Interface (API)	A software intermediary that allows two applications to talk to each other. APIs are an accessible way to extract and share data within and across organizations.
CSS (Cascading Style Sheets)	A stylesheet language used to create web pages. It describes how HTML elements are to be displayed on screen, paper, or in other media.
Graphical User Interface (GUI)	A form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators instead of text-based interfaces, typed command labels, or text navigation.
Git	A version control system that tracks changes in a set of source files. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.
GitHub	A platform for software development and version control using Git. It provides tools for collaboration, code review, and project management.
HTML (Hypertext Markup Language)	The standard markup language used to create web pages. It describes the structure of a web page and is used to embed text, images, videos, and other media.
JavaScript	A programming language used to create dynamic content on web pages. It allows for the implementation of complex features such as interactive forms, animations, and other elements that enhance user experience.
Rewards/Loyalty Program	A program offered by one vendor to entice customers to return, usually offering discounts or free products as incentives.
Rewards Credit Card	Credit cards that offer you some type of reward, typically cash back, points, or travel miles, for every dollar a customer spends. These rewards can be redeemed for various benefits.
WebCrawler	A program that systematically searches web pages for specific information. It is often used by search engines to index content from across the internet.

6. References

Horymski, C. (2024, April 24). *What is the average number of credit cards?*. Experian.

<https://www.experian.com/blogs/ask-experian/average-number-of-credit-cards-a-person-as/>

Deloitte. (n.d.). Taking a customer-centric approach to a data breach Insights from crisis response. Deloitte.

<https://www2.deloitte.com/content/dam/Deloitte/ca/Documents/risk/ca-Data-Breach-Customer-Centric-POV-EN-AODA.pdf>

Hsu, J., & Moss, E. (2022, August 10). How to Migrate Your Community to a Different Platform Successfully. Commsor.

<https://www.commsor.com/post/how-to-migrate-your-community-to-a-different-platform#:~:text=Switching%20community%20platforms%20is%20a,your%20team%20and%20key%20stakeholders>

Kumar, B. (2024, May 3). How To Make Money on YouTube in 2024: 7 Simple Strategies.

Shopify. <https://www.shopify.com/blog/198134793-how-to-make-money-on-youtube>

PayPal Honey. (n.d.). *PayPal Honey | Coupons and Cash Back*. PayPal US.

<https://www.paypal.com/us/digital-wallet/ways-to-pay/paypal-honey>

Rakuten. (n.d.). *Online Coupons & Cash Back: Shop 3,500+ Stores!*. Rakuten.

<https://www.rakuten.com/>

Rocket Money. (n.d.). *Learning Center*. Rocket Money. <https://www.rocketmoney.com/learn>

Soucy, P. (2024, July 3). *Best Credit Cards - June 2024*. NerdWallet.

<https://www.nerdwallet.com/the-best-credit-cards>