



March 27 2026

Morgan Nicholson, Faculty Mentor

Kyle Montgomery, Sponsor and Founder & CEO, Generous, Inc.

Zazel Turner

Christofer Vega

Josselin Retiguin Gutierrez

Technology Feasibility

Team: Generous, The Joy of Giving Project

Table of contents

- 1.Introduction
- 2.Technological Challenges
- 3.Technology Analysis
 - 3.1 Introduction
 - 3.2 Gift Card Redesign
 - 3.3 AI-Supported Search inside Gift Glimpse
- 4.Technology Integration
 - 4.1 Introduction
 - 4.2 System Diagram
 - 4.3 System overview
 - 4.4 Architecture integration
 - 4.5 Summary
- 5.Conclusion

1. Introduction:

Giving thoughtful gifts year-round sounds simple, but in reality it's something a lot of people silently struggle with. Families today are big and spread out, and it's very easy to lose track of who needs gifts and when. In the U.S., the average immediate family size is **3.1**, but once you factor in cousins, aunts, uncles, and extended relatives, that number grows fast. If one household has two or three kids, that alone creates multiple branches of relatives who all expect gifts for birthdays, holidays, graduations, and special events. Add in friends, coworkers, and partners, and the list becomes overwhelming. Trying to remember every date or every person's wants and needs quickly becomes a year-long challenge.

A lot of people try to manage this by last-minute shopping, giving gifts late, or asking spouses or parents of the recipient for help. And while there are gift-giving platforms out there, most of them still rely on the user doing most of the work. Amazon has wishlists, but they only help if the recipient has already filled one out. Target has registries, which are great for weddings and new babies but not everyday events like birthdays or anniversaries. Sites like GiftCards.com or CardCash just show long lists of gift cards with no guidance. Some people even use Google Calendar to remember dates, but that only helps with reminders — not choosing or sending a meaningful gift. None of these options help people pick a thoughtful gift *and* deliver it in a way that feels personal and trustworthy.

This is the gap our project is solving. We're redesigning the recipient experience so it reflects the thought and care the giver put into the gift. This includes updating the gift card display pages so the recipient actually feels the excitement of their upcoming gift and enhancing the Gift Glimpse so the recipient can smoothly and optionally adjust their preferences. Together, these improvements create a smoother, more personalized, and more meaningful experience — which is exactly what the sponsor needs for their updated vision.

The rest of this document breaks down the technical challenges, analyzes alternative solutions, and brings the architecture together into one cohesive system.

2. Technological Challenges:

- We will need a way to integrate AI-driven functionality into the Gift Glimpse feature that can accurately interpret user input, generate relevant gift recommendations, and support interactive modifications of the real time recommendation.
- We will need a way to design and implement a dynamic visually engaging gift card interface that supports real time updates and integrated image based content without negatively impacting system performance or load time

3. Technology Analysis:

3.1 Introduction:

As discussed in the previous section, the development of this platform introduces several key technological challenges that directly impact the overall user experience. While those challenges were identified at a high level, this section focuses on analyzing them in detail and determining the best approaches for implementation. The system relies heavily on how users interact with it, how information is presented, and how easily the platform can adapt to new features. One of the major areas of focus is the “Gift Glimpse” feature, which currently does not include any AI-driven functionality. This limits how helpful and interactive the system can be. By integrating an AI agent into this feature, the platform can move beyond static interactions and begin offering personalized suggestions while also allowing recipients to modify or exchange gifts through conversation. Together, these areas reflect the core challenges previously identified, and the following sections break each one down, explore possible solutions, and explain the reasoning behind the final choices.

3.2 Gift Card Redesign:

3.2.1 Introduction:

The current gift card interface displays results generated from user interaction with the AI, but lacks strong visual presentation and styling. Since gift selection is highly visual, improving the appearance and usability of gift cards is essential to enhancing user engagement and decision-making.

3.2.2 Desired Characteristics:

- Visual Appeal
 - Gift cards should include images and modern UI styling to attract users.
- Dynamic Updates
 - Cards should update instantly as the AI suggests new gifts.
- Adaptability
 - The design should be flexible for adding features (e.g. things like “swap gift” or “save”).
- Performance Efficiency
 - Rendering images and UI elements should not slow down the interface.
- Maintainability
 - UI components should be reusable and easy to modify.

3.2.3 Alternatives:

1. Amazon Product Advertising API (External Product Image Sourcing)

This approach uses the Amazon Product Advertising API to dynamically retrieve images for products that are already listed on our website. Instead of storing images manually, each product in our database is linked to an Amazon identifier (such as an ASIN), allowing our application to fetch up-to-date image URLs and display them in product cards at runtime. This reduces manual image management and ensures consistency with Amazon listings, but it requires handling API authentication, request limits, and compliance with Amazon’s usage policies.

2. React Component-Based Cards with CSS/Tailwind

This approach uses a component-based framework like React along with styling tools such as Tailwind CSS to dynamically render images within reusable card components. Images are typically stored in a backend, database, or external service and passed into components as data, allowing for features like responsive layouts, filtering, and lazy loading. This method is highly scalable and maintainable, making it ideal for applications that require a modern user interface and dynamic content updates.

3. Google Photos Integration (Manual and Cloud-Supported Image Management)

This approach uses Google Photos as a source for gathering product images, combined with manual downloading and integration into our website’s storage system. Since the project has

access to Google Cloud credits through our sponsor, images can be efficiently stored and managed using Google Cloud services after being selected. The workflow involves sourcing images from Google Photos, manually downloading and organizing them, optimizing them for web performance, and storing them in a way that supports responsive design (such as serving different image sizes for various viewport dimensions). This method offers flexibility and cost efficiency while still enabling optimized image delivery, though it requires more manual effort compared to fully automated solutions.

3.2.4 Analysis:

Visual Quality

Amazon API

- Score: 5
- Evidence: Based on Amazon Product Advertising API documentation, which provides standardized, high-resolution product images, ensuring consistent visual quality across product listings.

React + CSS

- Score: 5
- Evidence: Supported by modern frontend practices using React and Tailwind CSS, enabling responsive design, flexible layouts, and high-quality UI presentation.

Google Photos

- Score: 5
- Evidence: Manual image selection and optimization using tools from Google Cloud allow full control over image quality before deployment.

Adaptability

Amazon API

- Score: 3
- Evidence: Limited by predefined API structures, rate limits, and restricted data access as outlined in Amazon's official API guidelines.

React + CSS

- Score: 5
- Evidence: Component-based architecture (as described in React documentation) supports modularity, reusability, and easy integration with multiple data sources.

Google Photos

- Score: 3
- Evidence: Relies on manual workflows for downloading, organizing, and updating images, making large-scale changes less efficient.

Performance

Amazon API

- Score: 3

- Evidence: Dependent on external API calls, introducing network latency and potential delays, which are common limitations in API-based systems.

React + CSS

- Score: 5
- Evidence: Uses virtual DOM and supports optimization techniques like lazy loading, as documented in React performance guidelines.

Google Photos

- Score: 3
- Evidence: Performance depends on how well images are manually optimized and served, without fully automated delivery or CDN integration by default.

Maintainability

Amazon API

- Score: 3
- Evidence: Reduces manual image handling but introduces dependency on external services and requires ongoing API integration maintenance.

React + CSS

- Score: 5
- Evidence: Encourages modular, reusable code aligned with software engineering principles such as separation of concerns and maintainability.

Google Photos

- Score: 5
- Evidence: Simple workflow combined with cloud-backed storage via Google Cloud supports reliable and manageable image handling over time.

3.2.5 Chosen Approach:

Solution	Visual Quality	Adaptability	Performance	Maintability
Basic HTML	5	3	3	3
React + CSS	5	5	5	5
Google Photos	5	3	3	5

Decision:

React-based dynamic rendering is the best solution due to its performance and flexibility.

3.2.6 Proven Feasibility:

We will build a prototype interface that:

- Displays gift cards with images
- Updates cards dynamically during chat
- Includes improved styling and layout

We will test rendering speed, responsiveness, and user interaction with the redesigned UI.

3.3 AI-Supported Search inside Gift Glimpse:

3.3.1 Introduction:

The “Gift Glimpse” feature currently allows users to view and interact with gift options, but it does not include any AI-driven functionality. All suggestions and interactions are static or manually driven. To significantly enhance the user experience, this project proposes integrating an AI agent into Gift Glimpse.

This addition will enable both the gift sender and recipient to interact with the system conversationally. In particular, the recipient will be able to request modifications or exchanges for a gift, and the AI will generate alternative suggestions based on their preferences.

3.3.2 Desired Characteristics:

- Adaptability
 - The AI system should support multiple interaction types, including both gift selection and modification.
- Accuracy of Suggestions
 - Recommendations must be relevant to user preferences and context.
- Context Awareness
 - The system should consider previous inputs and the original gift when suggesting alternatives.
- Response Speed
 - The AI must respond quickly to maintain a smooth conversational experience.
- Ease of Integration
 - The solution should integrate cleanly with the existing Gift Glimpse system without requiring a complete redesign.

3.3.3 Alternatives:

1. Rule-Based System

This approach uses predefined rules and keyword matching to guide users toward suitable gift suggestions based on their inputs. Instead of relying on machine learning or external data, the system follows a fixed decision tree where specific keywords or phrases trigger predetermined responses and recommendations already configured in the platform. Each user query is scanned for recognizable patterns, allowing the application to map inputs to relevant gift categories or specific suggestions at runtime. This reduces the complexity of implementation and makes the system straightforward to maintain, but it requires manually updating the ruleset to accommodate new inputs, and it may fall short when handling varied, ambiguous, or unexpected user requests.

2. Traditional Recommendation Algorithm

This algorithm uses structured filtering techniques, such as category matching and user preference analysis, to surface relevant gift suggestions from the platform's existing catalog. Instead of generating responses dynamically, the system relies on organized data such as user behavior, predefined preferences, or product attributes already stored in the database, allowing the application to match users with gifts that align with their profile at runtime. This reduces the need for complex AI infrastructure and builds on proven methods widely adopted in e-commerce platforms, but it requires well-structured datasets and predefined logic to function effectively, and it may struggle to handle open-ended or nuanced requests that fall outside its established parameters.

3. Large Language Model (LLM) Integration

This alternative uses a Large Language Model to dynamically interpret user requests and generate personalized gift suggestions through natural language understanding. Instead of relying on fixed rules or structured datasets, the system processes conversational inputs in real time, allowing the application to understand context, handle ambiguous queries, and produce tailored recommendations fluidly at runtime. This reduces the rigidity seen in rule-based or algorithm-driven alternatives and delivers a significantly more intuitive and engaging user experience, but it requires careful infrastructure setup, API authentication, and ongoing attention to usage costs, response accuracy, and compliance with the model provider's usage policies.

3.3.4 Analysis:

Flexibility

Rule-Based System

- Score: 2
- Evidence: Limited to predefined rules and keyword matching, meaning it cannot handle unexpected or complex user input beyond what is explicitly programmed.

Recommendation System

- Score: 3
- Evidence: Uses structured filtering (e.g., categories, preferences), allowing some variation in outputs, but still constrained by predefined data relationships.

LLM Integration

- Score: 5
- Evidence: Large language models such as GPT models process natural language and generate responses dynamically, enabling flexible handling of open-ended and conversational input.

Adaptability

Rule-Based System

- Score: 2
- Evidence: Requires manual updates to rules for any new functionality, making it difficult to scale or extend to new use cases.

Recommendation System

- Score: 3
- Evidence: Can adapt by adjusting datasets or filtering logic, but still depends on structured inputs and predefined relationships.

LLM Integration

- Score: 5
- Evidence: LLMs can generalize across tasks and domains with minimal changes, as supported by modern AI system design and widespread use in conversational assistants.

Accuracy

Rule-Based System

- Score: 2
- Evidence: Accuracy is limited to exact rule matches and often fails when user input deviates from expected patterns.

Recommendation System

- Score: 3
- Evidence: Produces reasonably accurate results when sufficient structured data is available, as commonly seen in e-commerce recommendation engines.

LLM Integration

- Score: 5

- Evidence: Advanced models like those developed by OpenAI demonstrate strong contextual understanding and high-quality output generation in natural language tasks.

Development Effort

Rule-Based System

- Score: 5
- Evidence: Simple to implement using basic conditional logic and does not require complex infrastructure or external dependencies.

Recommendation System

- Score: 2
- Evidence: Requires structured datasets, algorithm design, and potential tuning, increasing implementation complexity and development time.

LLM Integration

- Score: 4
- Evidence: Moderate effort due to API integration and prompt design, but avoids building complex logic from scratch by leveraging existing LLM services.

3.3.5 Chosen Approach:

Solution	Flexibility	Adaptability	Accuracy	Effort
Rule-Based	2	2	2	5
Recommendation Algo.	3	3	3	2
LLM Integration	5	5	5	4

Decision:

Integrating a Large Language Model into the Gift Glimpse feature is the most effective solution. Unlike the current static system, this approach enables dynamic, conversational interactions and supports both gift generation and modification. Its high adaptability makes it especially suitable for extending the feature to new use cases without major redesign.

3.3.6 Proven Feasibility:

To validate this approach, we will develop a prototype that:

- Allows users to input natural language requests (e.g., “Can I swap this for something cheaper?”)
- Generates alternative gift suggestions using the AI model
- Updates the gift display dynamically based on AI responses

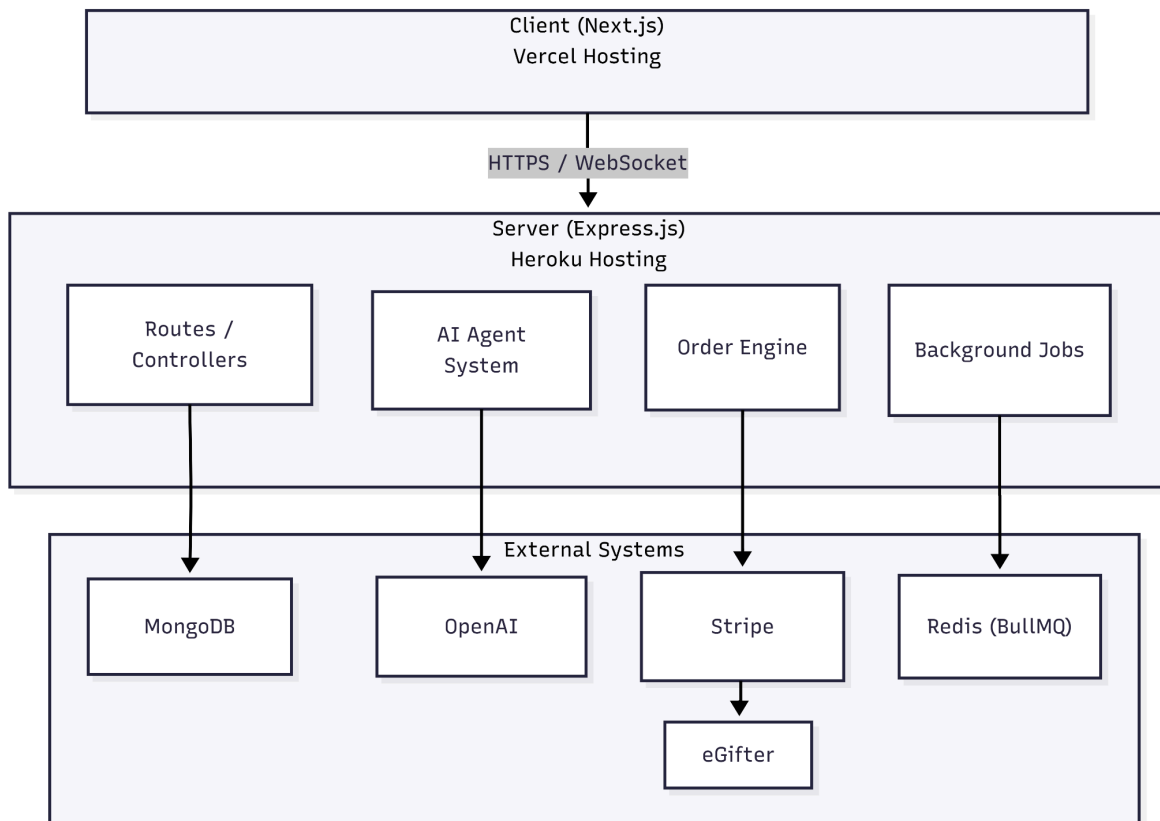
This prototype will demonstrate that AI integration can successfully transform Gift Glimpse from a static feature into an interactive, intelligent system.

4. Technology Integration:

4.1 Introduction:

Now that we have explained the approach for our solutions, which outlined what technologies we are going to be using and why, we will dive deeper into how these solutions fit into the architecture of this program. It will present how the technical solutions identified in the analysis phase combine into a unified, headless system architecture. The goal is to show how the platform's components work together to satisfy the product requirements. The sponsor provided system diagram serves as the structural reference for the integration.

4.2 Current System Diagram:



4.3 System Overview:

This platform follows a headless architecture where the Server contains all the business logic, data integrity and operational workflows. The Client applications, using next.js and for future differing platforms, interact with this system through the API contacts. This separation makes sure that when new platforms are introduced there is no need to re-implement or develop core logic, also ensuring that all the clients will behave consistently.

The system diagram above displays three major layers.

- Client layer: UI focused applications responsible for user interactions and rendering
- Server Layer: Express based backend that is hosted on heroku. This is where all the AI agent system, order engine, controller and background jobs are orchestrated.
- External Layer: This is where the third party systems such as mongoDB, OpenAI, stripe and redis all live.

4.4 Architecture Integration

Client Layer

The client application will communicate with the server through HTTPS and will send real-time updates through WebSockets. The Client Layer will delegate all decision making to the Server Layer, this keeps the consistency throughout the differing platforms. This will align with the sponsor's designated headless structure.

Server Layer

The server Layer will orchestrate the platform behavior integrating several systems such as:

- Routes and controllers: handle incoming requests, validating input, and routing actions.
- AI Agent System: interprets user intent, executes tool-based workflows and interacts with language models.
- Order Engine: manages lifecycle of gift orders.
- Background Jobs: Redis and BullMQ are used to schedule and execute tasks such as delayed payments and gift deliveries.

External Layer

The server integrates with multiple external systems such as:

- MongoDB: where Data is stored.
- Redis and BullMQ: used for caching.
- Stripe: handles the payment authorization.
- eGifter: provides the gift card fulfillment.
- OpenAI/Claude: supports AI driven conversation flows.
- *Google Photos/Catalog system*: serve as *Potential* sources for the product images.

4.5 Summary

This systems integrated architecture forms a system in which each component plays a specific defined role. The Client will focus on the presentation, the Server will enforce the business logic, and the External Services will provide the specialized capabilities we need. This design supports the functional requirements while still enabling the expansion and maintainability of this product.

5. Conclusion:

This project started with a simple but important problem: people want to give thoughtful gifts, but the current receiver side Generous platform doesn't fully support the clarity, trust, or modern experience needed to make that happen. Throughout this document, we looked closely at the areas holding the receiver experience back—the gift card display page and the AI-supported search—and focused on what users actually expect when they open a digital gift.

From there, we evaluated different technical approaches and chose solutions that make sense for the product, the timeline, and the sponsor's priorities. Our improvements center on trust and convenience: a consistent and theme-driven gift card display page and an optional AI assistant that supports the user without adding extra tasking. Both of these changes directly address the gaps we identified earlier in the platform.

Finally, we brought all of these pieces together into one unified architecture. The Client, Server, and external services each play a specific role, but they stay connected through shared data, consistent theming, and a clean flow from the initial email all the way to the final redemption of the gift. The result is a system that feels more personal, more reliable, and more aligned with the emotional experience the sponsor wants to deliver.

With this foundation in place, we're ready to move into implementation. Our next steps are to build out the production-ready UI improvements and deliver the research-driven features that will make the Generous platform feel trustworthy, modern, and genuinely helpful for both gift givers and receivers. This feasibility study shows that the work ahead is not only possible—it's supported by the technology stack, grounded in user needs, and structured in a way that sets the project up for success.