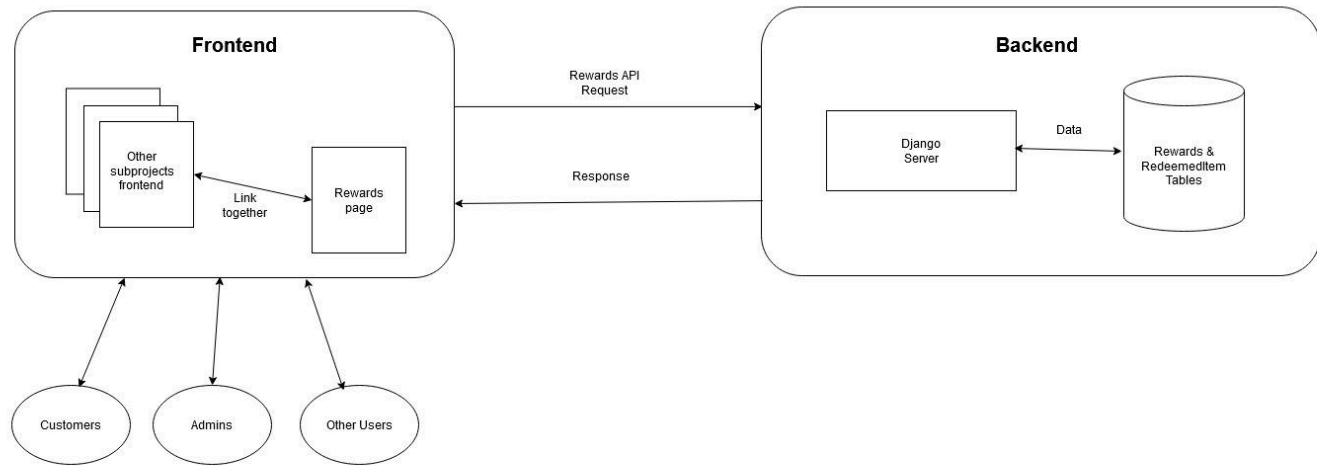


Rewards & Loyalty Program Architecture

Rewards sub-project Architecture



We are employing a basic web-app architecture where users would interact with the app through the front-end server which communicates with the back-end server via API calls in order to get data or perform some function that changes the database.

We have our backend set-up in a django server. For all points earned/redeemed, this information is stored in the Rewards table and all the redeemed items with item-specific information such as coupon codes are in the RedeemedItem table.

The server has some api calls set up for all our functionality required for our rewards page along with functionality required for the other sub-projects and the overall functionality required for the loyalty/reward points sub-project.

The front-end would be on a different server and would communicate with the backend server through these api calls.

Some more information on system design is given in the next pages.

System Design

Back-end specifications

The backend functionality will be implemented through the following API calls split by the features which they implement. Each endpoint has a prefix of 'rewards-api/'.

Point accumulation:

API endpoint	Input (through POST)	Output
earn-points/	user_id: int points_earned: int reward_description: string	201 created status. rewards table entry created.
purchase/	user_id: int reward_description: string amount: double	201 created status with a message of points earned, calculated based on price of item (amount) purchased. rewards table entry created.

Rewards Catalog:

API endpoint	Input (through POST)	Output
redeem-coupon/	user_id: int points_redeemed: int reward_description: string coupon_amount: double	201 created status with a message of coupon code for coupon. active rewards table entry created. redeemedItem table entry created.
redeem-donation/	user_id: int points_redeemed: int reward_description: string donation_amount: double	201 created status. active rewards table entry created. redeemedItem table entry created.

Redemption Tracking:

We are putting redemption tracking in the rewards catalog page. Along with coupon or donation redeem options, they will also see their total points and redemption history, as well as the status of each item (is_active attribute on rewards table).

API endpoint	Input (through POST)	Output
get-points/	user_id: int	200 OK status. total_points: int total_points_earned: int

		total_points_redeemed: int
get-redemptions/	user_id: int	200 OK status. redemption_history: list of { is_active: boolean redeemed_date: DATETIME redeemed_item: { item_code: string item_description: string amount: double } }

Bonus Points for Engagement:

Along with feedback (below), we also have the earn-points that can be used for any general-purpose loyalty points opportunities. So if the user attends some event, the front-end server can simply call earn-points/ for that user and specify an amount of points to reward them.

API endpoint	Input (through POST)	Output
submit-feedback/	user_id: int reward_description: string	201 created status. rewards table entry created. NOTE: each feedback review submitted will earn the user a flat 10 loyalty points

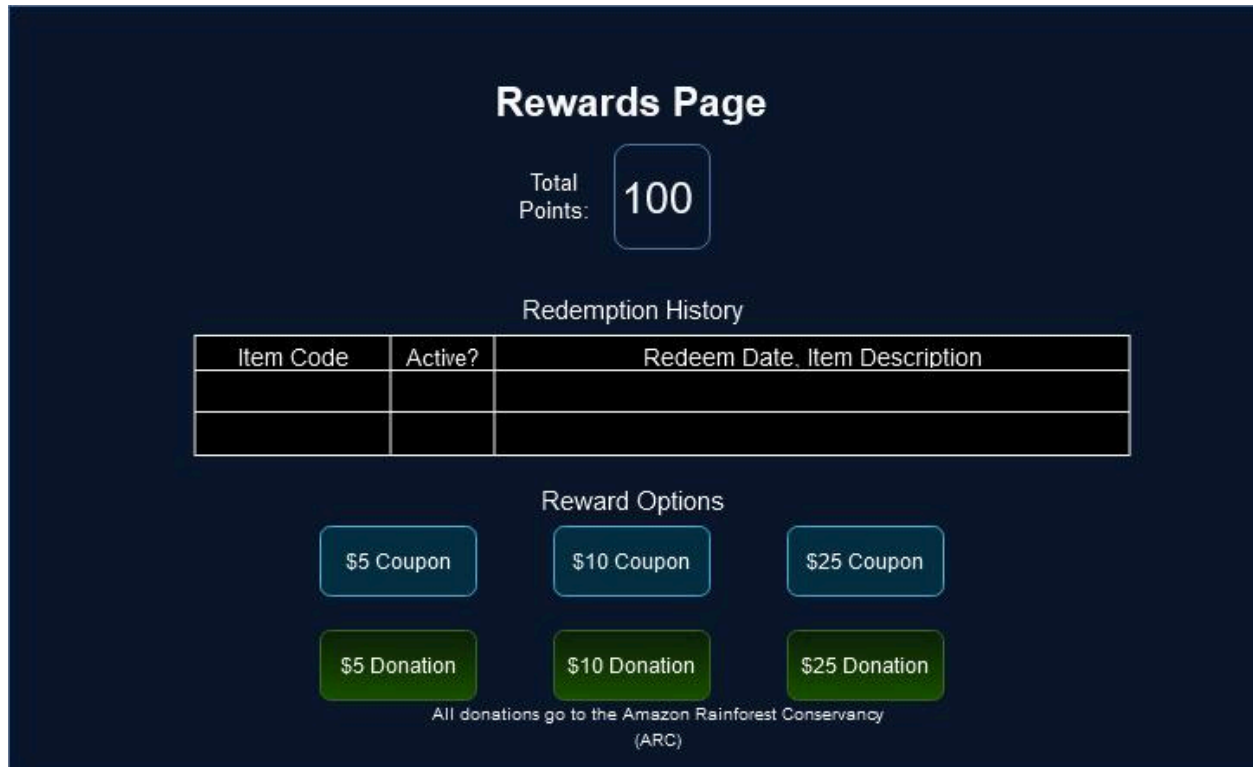
Other API calls:

The admin dashboard and user profile team will also be using our backend for their own purposes, we have just 1 API endpoint for both sub-projects. This returns the total points of that user as well as all reward table entries.

API endpoint	Input (through POST)	Output
get-user-info/	user_id: int	200 OK status. total_points: int total_points_earned: int total_points_redeemed: int reward_history: list of { points_earned: int points_redeemed: int earned_date: DATETIME redeemed_date: DATETIME reward_description: string }

Front-end specifications

Simple UI mock-up:



This rewards page will be accessible through the user profile. It will combine the rewards catalog and the redemption tracking requirements, so we only have the one webpage.

API calls used:

rewards-api/get-points/ = To get Total Points

rewards-api/get-redemptions/ = To get Redemption History

rewards-api/redeem-coupon/ = To redeem a specific coupon option

- (Returned coupon code is presented to user via an on-screen alert)

rewards-api/get-redemptions/ = To redeem a specific donation option