Logo

Description automatically generated

***STAEM***

**Overview**

**Staem** is a platform for **worldwide gaming distribution**, **in-game item transaction marketplace**, and **communities** to share your gaming experience.

The project manager, **Abenbyy** wants you to create a website for **Staem.** Since the platform is intended for global distribution, the website should be able to **support non-latin characters** such as Mandarin, Japanese, Russian, etc.

Upon development, **Abenbyy** expects you to **know what you are using**, and **why you are using it. Abenbyy** wants you to use **Git** version control so both you and **Abenbyy** can keep track on your work by seeing your commits every day. You could use **GitHub** or **GitLab** for the git repository manager. Since this is a confidential project make sure your repository is set to **private** so nobody can copy and steal your work. **Abenbyy** expects you to uphold your **integrity** so make sure every time you’re about to present your work to **Abenbyy**, you’re presenting what you had done until the **last commit before the deadline.**

In this project, you are asked to use **GO** language as the backend language to create **GraphQL API,** to make sure it is a GraphQL API, your backend must be accessible by only 1 endpoint. You are also required to use **Angular** as the frontend framework to develop your web application. The language that you must use for the Angular framework is **TypeScript.** For the database you must use **PostgreSQL.** You are **free to create and design the database** in any section, but you must have a **strong logical reason why**. To make sure the application works, both your frontend and backend must be accessible by using **public IP/domain.**

There are some factors that you should notice on development such as **performance, security, design, and tools understanding.**

In the performance factor, you are asked to use **Redis** to cache the result of any API data you consume. Use **RxJS** JavaScript library to help you deal with asynchronous data transfers.

In the design factor, the web design must be responsive with 3 breakpoints, **desktop, tablet,** and **mobile.** You also asked to make your application support **progressive web applications** to deliver amazing user experience.

Since **Abenbyy** notices that so many junior developers don’t have a strong foundation in **CSS, Abenbyy** hopes your **CSS** skills will **increase significantly** after you **finish** this project. With the intention to improve your **CSS skill** you are prohibited to use any **CSS** **library or UI component libraries** such as Bootstrap, Material UI, Prime, etc. **Abenbyy** wants to share the wonderful feeling of using **SASS/SCSS** styling language, so you are asked to use it.

Other than **UI** and **Design,** please consider the **UX** of the application, because **the experience** is what makes customers loyal to the web, **not the design.**

In security factor, you are asked to use **OAuth2** for login authentication process.You may use any authorization server as long as it’s free. If you found any bugs in any technology you are using, report bugs to the official website and write your documentation what bugs you had found.

In tools understanding factor, you are asked to just simply not just use the tools, but **understand** *what*, *why*, *how* and anything that *related* to the tools you are using. For example: Do the tools implement any design pattern or infrastructure patternthat you maybe could add to your project.

While working on this project, remember what’s the main functionality of the web, focus on **learning something new**, find some motivation when you’re unmotivated, rest when you are tired, work smartly, be creative, and have fun.

**Project Requirements**

1. **Header**

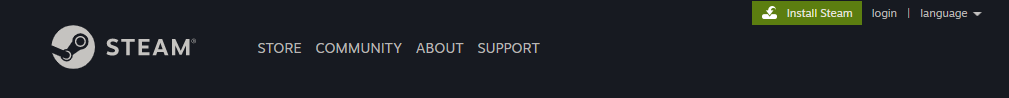
****

Figure 1. Guest Header

* *Guest Page*
* Display the website **logo**
* Display **navigation bar menus** available to the website.
* Display **login** button and redirect users to **Login Page** when clicked
* Display **language dropdown**, change the website language based on the currently selected language

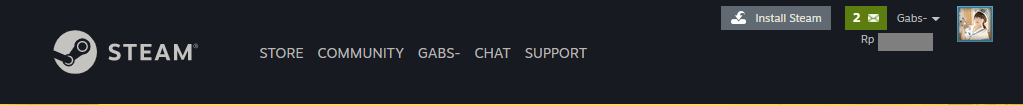
****

Figure 2. User Header

* *User Page*
* Display **notifications** to notify user about:
* New comments on user post
* New items in user inventory
* New friend requests from other users
* New gifts that user received from other users
* New unread messages from other users
* Display **user display name**
* Display **user current wallet amount**
* Display **user profile picture** (small)

1. **Footer**

* Provide **10 useful site links** to access site pages

1. **Side Bar**

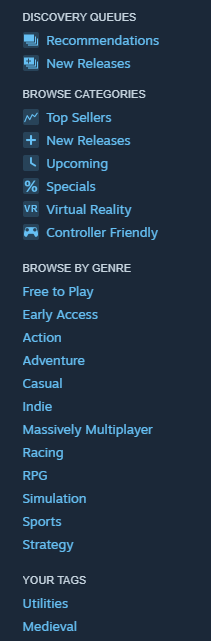
****

Figure 3. Side Bar

* Display **Discovery Queue sub-menu**, when clicked redirect to **Discovery Queue Page**, based on **Recommendations** or **New Releases**
* Display **Browse Categories sub-menu**, when clicked redirect to **Search Game Result Page**, based on **Category**
* Display **Browse by Genre sub-menu**, when clicked redirect to **Search Game Result Page**, based on **Gere**
* Display **current users most viewed genre** based on the **games that the user viewed**

1. **Home Page**

* *Guest Page*
* Display **Search Bar** to search games:
* When user types the query, **display top 5 games related to the user input**

(ex. When user types “do”, display games containing the word “do”, such as “Dota 2”, “Don’t Starve Together”, “DOOM Eternal”, etc.)

* When user clicks **search button**, **find games that the user has searched** and redirect to **Search Game Page**
* Display **Featured and Recommended** games, these are the games with the **highest number of gameplay hours in the past week**
* Display **Special Offers** games, these are the games that are **currently on sale**
* Display **Community Recommended** games, these are the games that has **most positive reviews on the past week**
* Display **Streams**, these **are streams from other** **users that are currently playing**
* Display Games for each category
* The categories are **New and Trending (recently published), Top Sellers (purchased the most in a week),** and **Specials (on sale > 50% only)**
* When user hovers the game, **display the game summary** on the right section

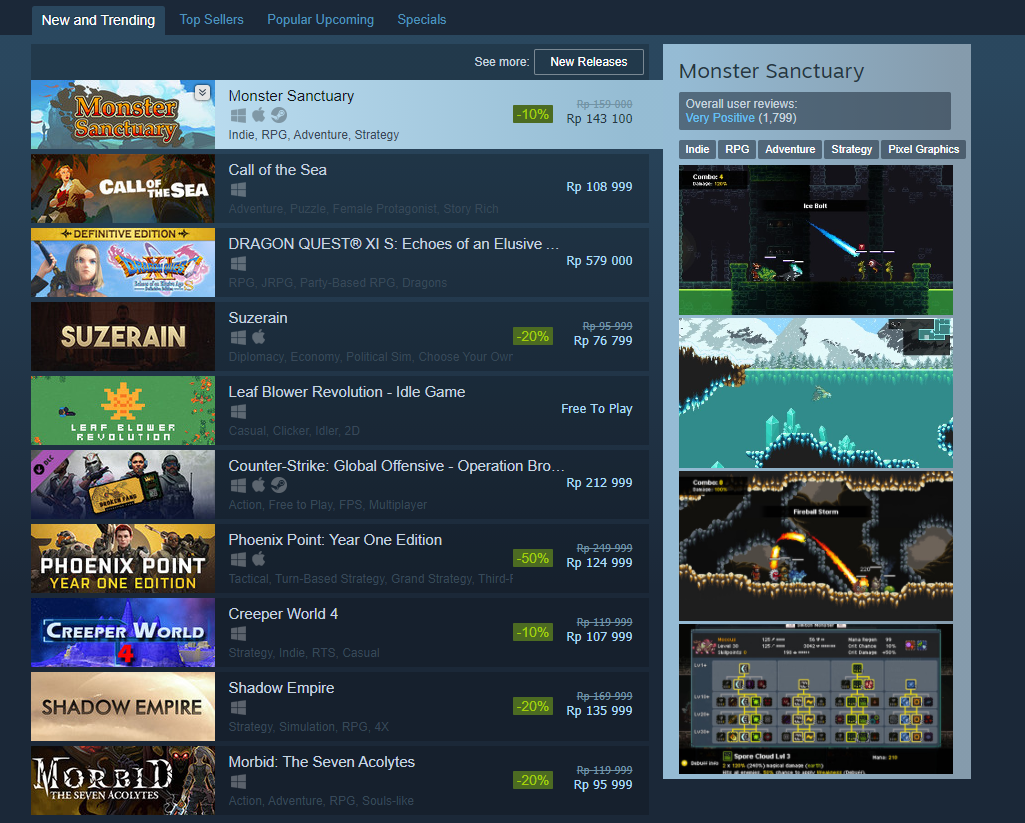
****

Figure 4. Games Based on Categories

* *User Page*
* Display “**Discovery Queue”** preview, when clicked redirect to **Discovery Queue Page** (by recommendation)
* Display “**Trending Among Friends’,** these are the games that are **played the most by the users’ friends**.

1. **Search Game Page**

* Display **search result from user query**, use **infinite scroll** to display the data. **Add 10 data every time the infinite scroll is triggered**
* When a **game is hovered**, display the **game’s overview**
* Display filters to narrow the search result, available filters are:
* Filter by price (use slider)
* Filter by tag (genre)
* Filter by category (top seller, new releases, specials)

1. **Game Detail Page**

* *Guest Page*
* If the game contains **mature content** such as violence, nudity, etc. (from the games tag), prompt the guest’s birthdate. If the guest’s age is less than 17, redirect to homepage. **The birthdate will not be saved**
* Display **breadcrumbs** to the site link
* Display a **slideshow** containing **videos and images** about the game
* Display **details** about the game, such as description, developer, publisher, price, system requirements
* Display **maps** containing the **number of players who owns the game in each country** (minimum of 5 countries)
* Display **add to cart button**, when clicked **add the game into cart**

****

Figure . Review Section

* Display reviews, the reviews are split into 2 categories:
* **Most Helpful Reviews**, these are reviews that are upvoted the most in the past month (30 days)
* **Recently Posted Reviews**, these are reviews that are most recently posted
* *User Page*
* Display **add to wishlist button**, when clicked add the game into the **users’ wishlist**
* If user **owns the game**, **display a form to post a review**
* User can **upvote** or **downvote** each review

1. **Cart Page**

* Display **items in cart**, display details such as game name, image, price
* Display **remove button**, when clicked **display confirmation dialog** (modal)to minimize user error,if user confirms **remove the item from the cart**
* Display **subtotal for items in cart**
* Display **Purchase for myself** and **Purchase as a gift button,** when clicked redirect to **Checkout Page**

1. **Checkout Page**

* *Guest Page*
* If the user is not authenticated yet, redirect to Login Page
* *User Page*
* *Purchase for myself*
* If the user has enough wallet, display confirmation form on the purchase
* Display change payment method example, when clicked display payment form
* If the user doesn’t have enough wallet, display payment form

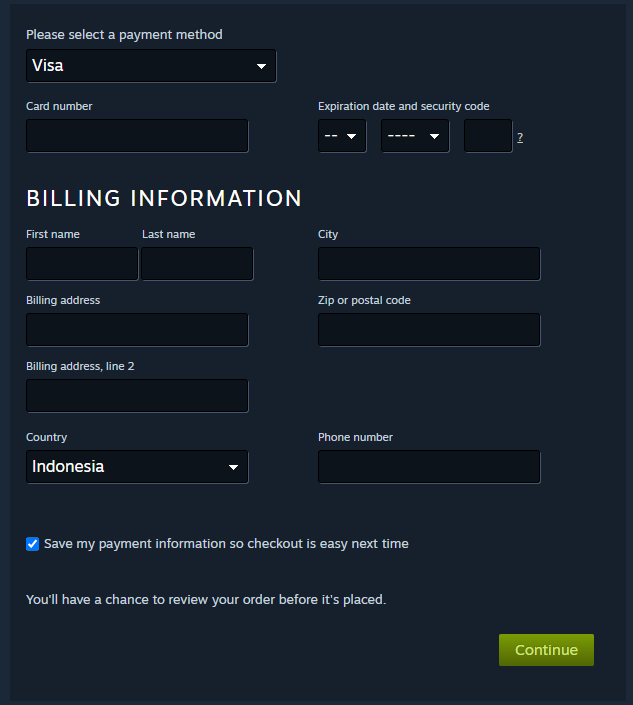


Figure 6. Payment Form

* *Purchase as a gift*
* Display **form to choose the gift’s recipient**
* Display **form to add note to the gift**, containing the recipient first name, message, sentiment, and signature
* Send a **purchase receipt** on the user’s **registered email** after a successful transaction

1. **Login Page**

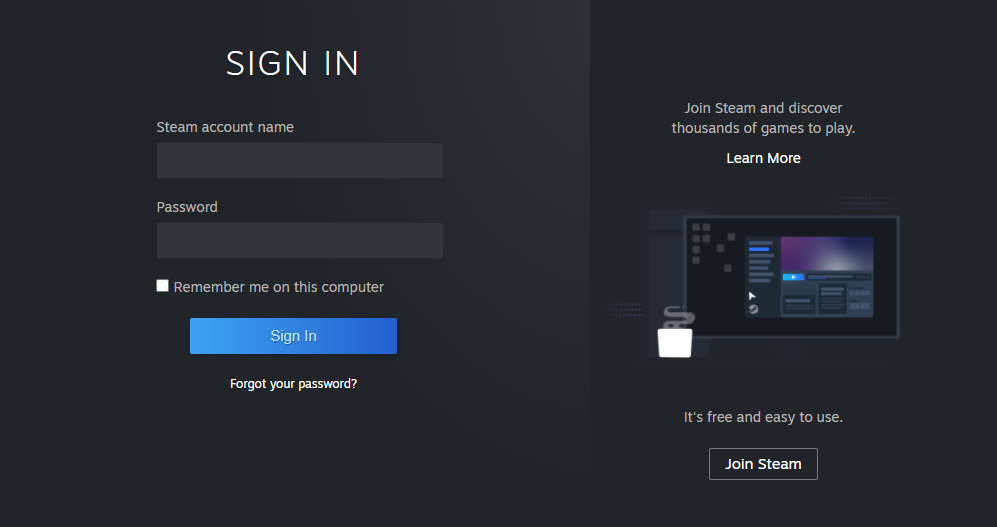
****

Figure 7. Login Page

* Prompt user to **input account name and password**, if user click **Sign In** button, validate user credentials
* If credentials are **invalid**, **display error message**
* If credentials are **valid**, **redirect to Home Page**
* If credentials are **valid**, but the account is currently **suspended**, **display a modal to notify the user that their account is suspended** and **provide a form** so that user can **request to admin to unsuspend** the account along with the reasoning.
* If user checks remember me, **save user credentials** locally
* Display **Join Steam button**, when clicked redirect to **Register Page**

1. **Register Page**

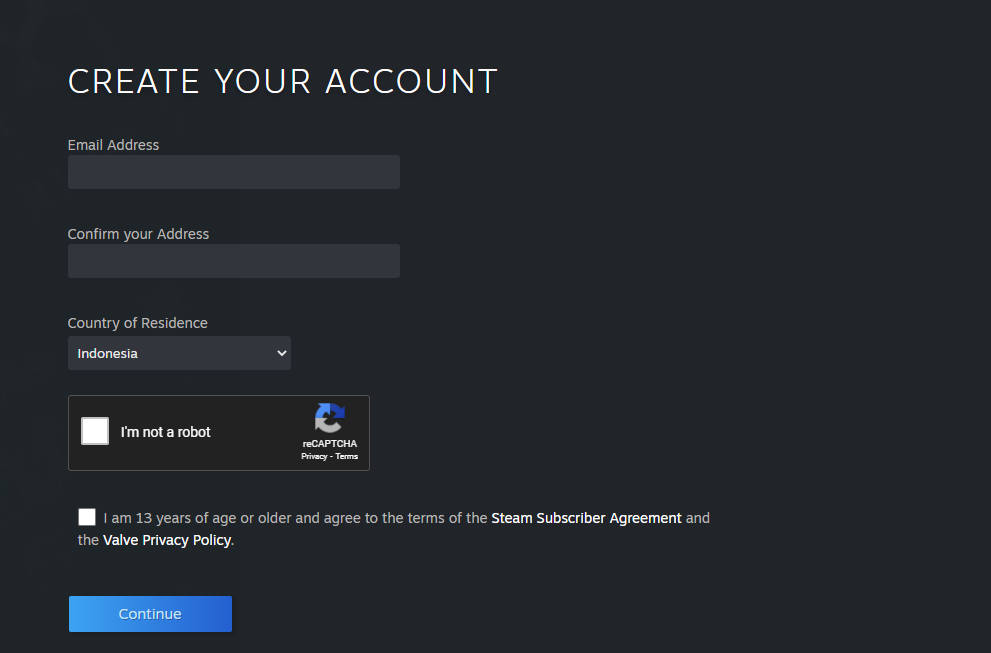


Figure 8. Register Page

* Prompt user to **input email, confirm address, country of residence, and agree to terms**
* Use **reCAPTCHA security service** to validate user registration
* When user click **continue button,** continue user registration:
* **Generate an OTP** code and **send it to users’ email**, OTP code should be **an alphanumeric string with 5 characters**
* Prompt user to input:
* Account name
* Password
* Confirm password
* OTP code from email
* When user finishes input, create an account based on inputted credentials

1. **Wishlist Page**

* Display **games added to users’ wishlist**
* When **games on wishlist is on sale**, user will be **notified by email.** Use observer design pattern on designing your code.

1. **Market Page**

* Display **Market Item List** and order them by the **number of transactions descending**. Display the data using **pagination**

1. **Market Item Detail Page**

* Display Item detail such as image, game, and description
* Display live information on sales and purchases, the data will update every 5 seconds:
* Display **Buy Button**, when clicked show create listing modal to list users’ request
* Display **Sell Button**, when clicked show list item modal to sell items in inventory
* Display **Sell Listings**, if user create a sell listing, it will be displayed here
* Display **Buy Listings**, if user create a buy listing, it will be displayed here
* Display **graphs to visualize the item sales** in the past month
* Display **Recent Activity**, it will use **web socket** to display information. The information to be displayed are:
* User **buy** item from other user
* User **create sale** list for the item
* User **create buy** list for the item
* When user **successfully sells an item**, the user wallet will be **added** (with 10% commission fee), and a **transaction receipt will be sent via email**
* When user **successfully buys an item**, the user wallet will be **deducted**, and a **purchase receipt will be sent via email**

1. **Top Up Wallet Page**

* Display **form to prompt** wallet code
* Display **redeem button,** when clicked validate if the code is valid, and top up the wallet according to the code value

1. **Profile**

* *All Profile*
* Display **profile picture** (large)
* Display **name, level,** and **summary**
* Display **featured badge** selected by the user
* Display **games** owned by the user
* Display **friends**, along with their **status** (currently playing, online, offline)
* When **friends profile** is hovered, show **mini profile**

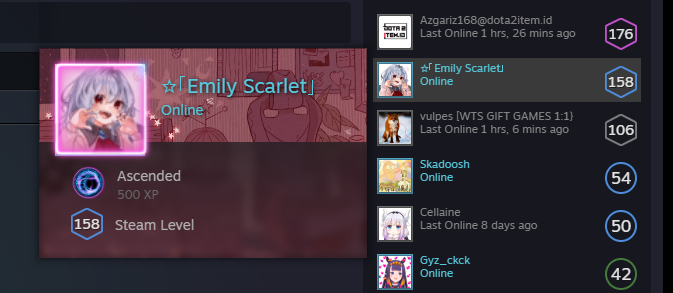


Figure 9. Mini Profile Example

* Display **comments**, these are **comments that other user post** on current user profile
* Display **recent activity** using **pagination** (display 10 item per page) such as:
* Recent Gameplays
* Community Posts
* Reviews
* *If User Profile*
* Display **edit profile button**, when clicked redirect to **Edit Profile Page**
* *If Other User Profile*
* If the user is **friends with current user,** display **message button**, when clicked, **start a chat with the visited user**.
* If the user is **not friends yet with current user**, display **add friend button**, when clicked **send a friend request to the user**
* Display a **report button**, when clicked **prompt a modal to fill out details about the report.**
* If a profile is reported **more than 5 times in the last week**, the account will be **automatically suspended**.

1. **Edit Profile Page**

* User can **view** and **edit**:
* Display name
* Real name
* Custom URL
* Country
* Summary
* Avatar
* Avatar frame
* Profile background
* Mini profile background
* Theme
* Featured Badge

1. **Inventory**

* Display **in game items**, each game has its own tab
* Display **user owned items**, display it using **simple pagination**
* Display **search bar,** when user types **filter the items on current category**
* When **item is clicked**, display **item summary**
* Display **sell button**, when clicked show create listing modal
* Display **item image, name, and game**
* Display **graph to visualize sale prices** in the past month
* Display **form for user to input either the amount that will be received or the amount that the buyer pays**, the amount received will be the amount that buyer pays – 10%

Ex: If the listed price is Rp. 10.000, the seller will receive only Rp. 9.000

* Display **sell button**, when clicked c**reate listing based on the inputted price**

1. **Friend Page**

* Display **search bar**, when user types filter friend list
* Display **friend list**, divide them **based on their status** (online/offline)
* Display report button, when clicked prompt a modal to fill out report details such as,
* Display **pending invites:**
* Received invites:
* Display **requesting user image, name,** and **level**
* Display **accept button**, when clicked **add user to friend list**
* Display **decline button**, when clicked **remove request from both current user and requesting user**
* Display **ignore button**, when clicked **remove request from the current user only** (the requesting user will still show pending)
* Sent invites
* Display outgoing friend requests
* Display add friend button, display form:
* Display **Friend Code**, when other users enter this code, they will **send friend request to the current user**
* Display **Friend Code form**, when user enters code, validate the code. **Send friend request to that user if friend code is valid.**

1. **Badges Page**

* User can collect trading cards for each game, when the collection is complete, players can trade the cards for a badge and gain EXP for their account level

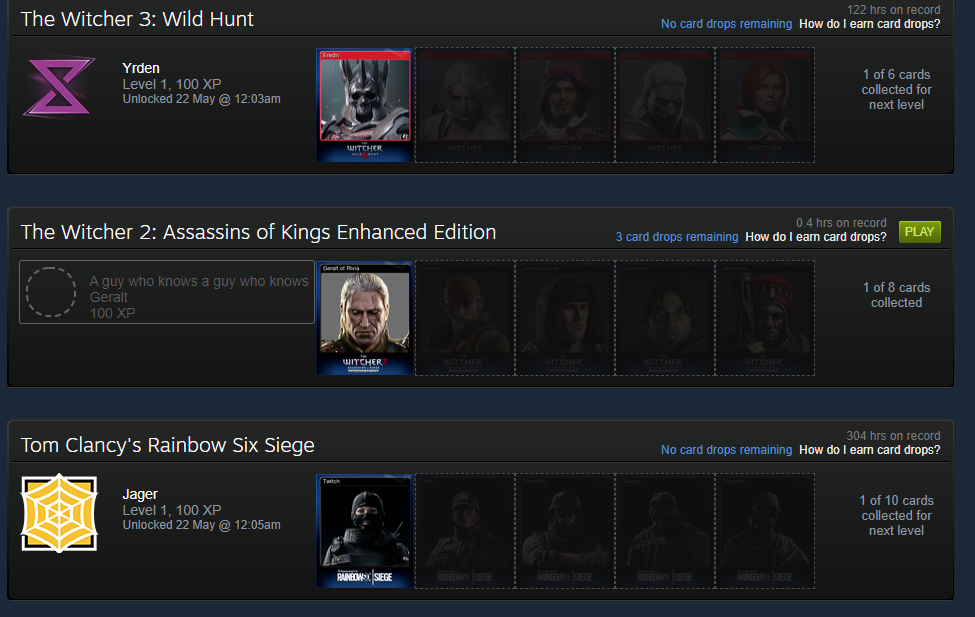


Figure 10. Badges Page

* User can sell and buy these cards from the marketplace, each time the collection is complete, the badge will level up and the user account will gain more EXP (the amount of exp is up to you to decide)

1. **Community Page**

* Display posts from users in the **Staem** community, the tabs are as follow:
* Images + Videos
* Display the posted image or video and description
* Display like and dislike button, when clicked, the post likes will increase or decrease respectively
* When clicked, display details
* Display poster’s image, and display name
* Display reviews along with reviewer’s profile, display data using pagination (10 items per page)
* Display add comment form, when posted, add review to the post
* Reviews
* Display the review game, recommend or not, and the description
* Ask whether if the review is helpful or not, if yes, add the number of users that find the review helpful
* When clicked, display details
* Display poster’s image, and display name
* Display reviews along with reviewer’s profile, display data using pagination (10 items per page)
* Display add comment form, when posted, add review to the post
* Discussions

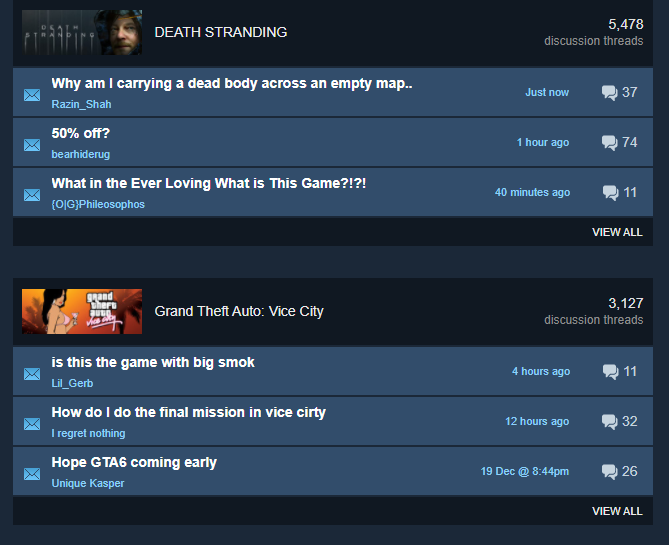
****

Figure 11. Discussions Tab

* Display discussions grouped by game, display top 3 forums based on most recent discussion

User can search the game, when search button clicked, display game’s discussions according to user search. (ex. When user types “do”, display game discussions containing the word “do”, such as “Dota 2”, “Don’t Starve Together”, “DOOM Eternal”, etc.)

* Display **create discussion** button, when clicked, prompt user to fill out discussions **title, game**, and **description**. After user is finished, **create a new discussion thread on the selected game.**
* When discussion is clicked, display discussion details
* Display discussion’s poster profile, title, and content
* Display replies to the discussion
* Display reply form, when posted, add a reply to the discussion

1. **Discovery Queue Page**

* *By Recommendation*
* Display **10 recommended games**. The games are **personalized based on the current game that the user owns.**
* Use **Pearson Correlation** or **Cosine Similarity** Algorithm, to **predict games that the user may like**
* *By New Releases*
* Display **10 newly released games**. The games are based on **most recent release date**

1. **Broadcast Page**

* Display **available stream that are currently ongoing on the website**
* When clicked**, play the stream on the website**
* **You don’t need to have a live chat feature, just livestream is fine**

1. **Stream Page**

* User can **stream their screens**, and **other viewers can view the user** stream, you may use third party software to capture screen such as **OBS**, and use video broadcast library such as **WebRTC**

1. **Chat Page**

* User can **chat with their friends**, use **web socket** to enable **real-time chat**
* User can **send emoticons**,and **stickers** in the chat
* User can **send** and **download** **files** and images in the chat

1. **Point Page**

* To attract even more customers, a point feature is added
* Each time user spends money through **buying games** (not marketplace) they will earn points (example every Rp. 15000 spent, they will get 100 points), refer to <https://store.steampowered.com/points/howitworks#Title> for a more deeper understanding on how it works
* User can exchange points to numerous items, such as **profile background**, **avatar border, animated avatars**, **chat stickers**, and **mini profile backgrounds**.

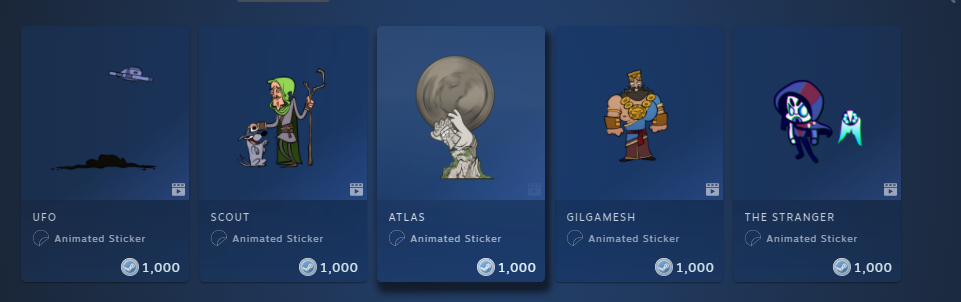


Figure 12. Animated Stickers



Figure 13. Animated Avatars

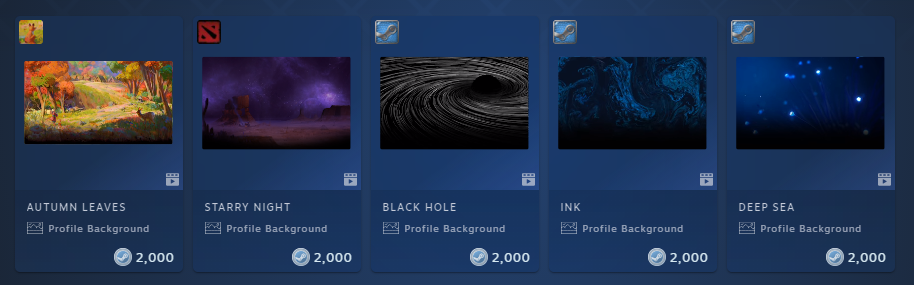


Figure 14. Profile Backgrounds

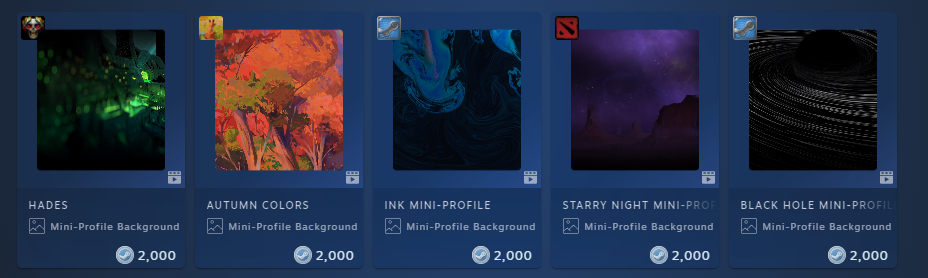


Figure 15. Mini Profile Backgrounds

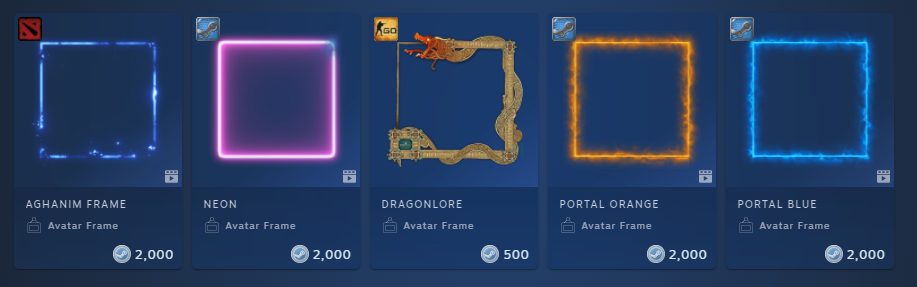


Figure 16. Avatar Frames

1. **Admin Login**

* Admin will not login from normal login page, admins will have to log in from a different page
* **Create a page for admin login**

1. **Admin Manage Game**

* Authorize admin before entering the page
* Admin can view all available games, display data using pagination
* Admin can insert, update, and delete:
* Game title
* Game description
* Game price
* Game banner
* Game slideshow (video and images)
* Game tags (such as Horror, Adventure, Indie)
* Game system requirements

1. **Admin Manage Promo/Discount**

* Authorize admin before entering the page
* Admin can view all available promo, display promo data using pagination
* Admin can insert, update, and delete:
* Promo discount
* Promo duration

1. **Admin Manage User**

* Authorize admin before entering the page
* Admin can view all available users, display data using pagination
* Admin can view user data such as profile, name, summary
* Admin can view list of reports on a user’s account, and who reported it
* Admin can suspend the user, when suspended, user cannot login and must fill out a request to unsuspend their accounts
* Admin can view users requests to unsuspend their account, and either **approve** or **deny** unsuspension.