Documentation for ECM2434

LoveBugs - Glapped

March 2025

Contents

1 Problem Statement

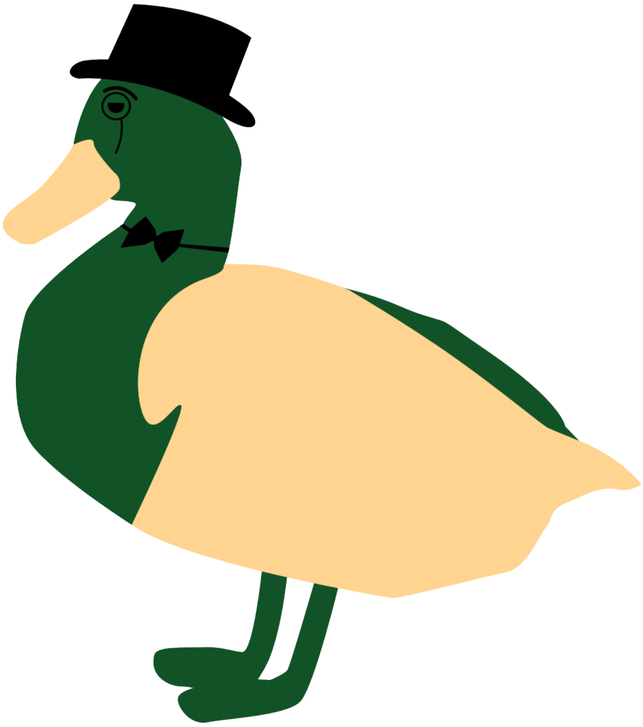
The requirements for this project are to develop a web application to promote sustainability through gamification on the Exeter University campus. To this end, the project is an app that allows users to trade items of clothing in return for points. These points are earned by selling items, with users bidding on an item with their points, and a flat rate of points being earned for the poster per item. These can then be exchanged in clothes swaps, encouraging users to engage with both sides of the clothes swap. The amount of estimated carbon dioxide and water the user saves during the clothes swap process is recorded, this will be public information on their account page and displayed in public leaderboards to encourage gamification.

Promoting sustainability in clothing is important. The rise of fast fashion has led to wasteful production of items meant to be worn only a few times. As a way to combat this, we are developing software that aims to promote reuse of unwanted clothes as well as the longevity of items through a second-hand clothing store-front.

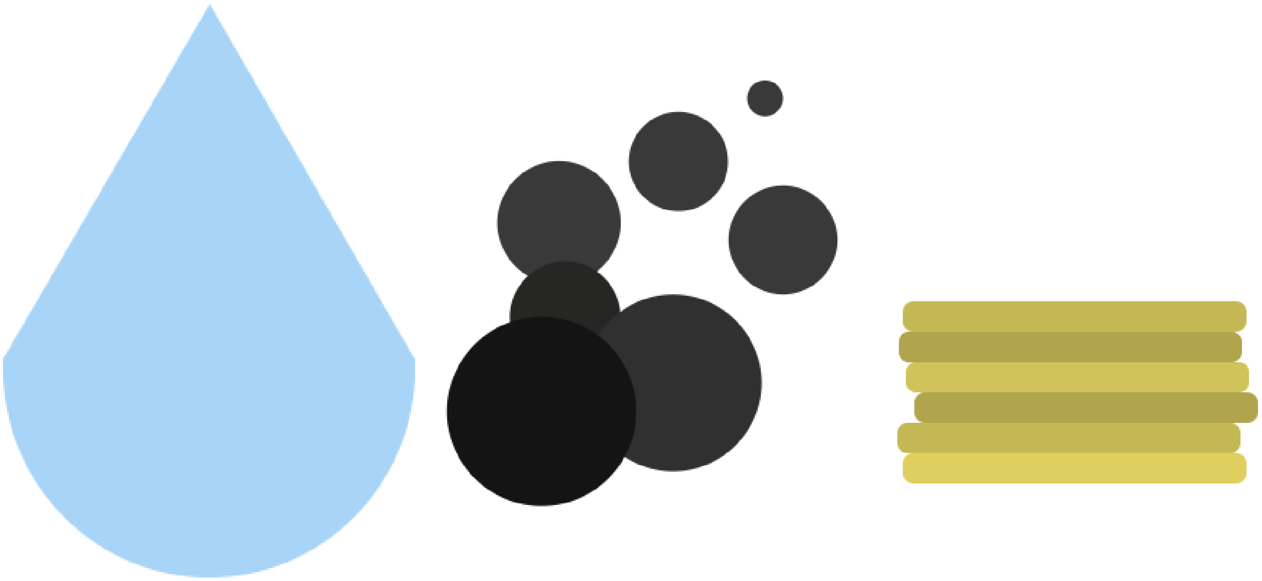
2 Mock-ups

- Ui/Ux Mock-ups - Some planned features will be out of scope for the first sprint, but are still important to consider and plan for.

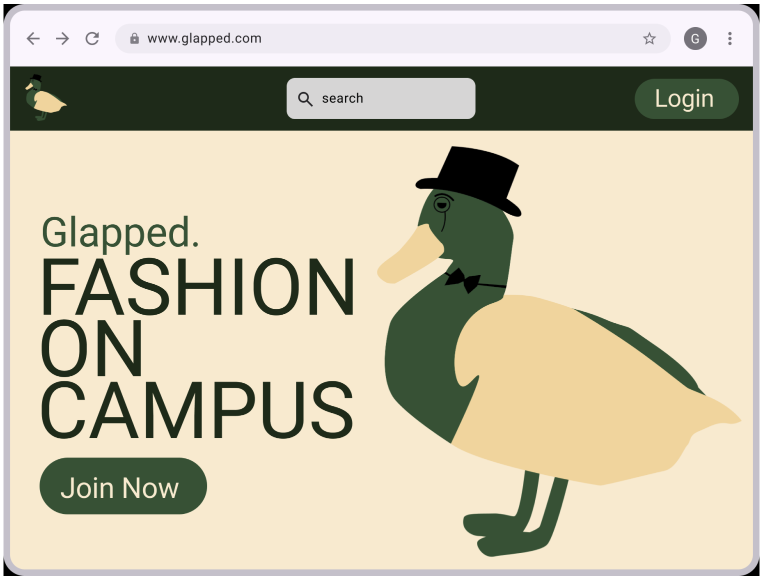
- Logo - Our logo aims to link our application with campus, with ducks being a fairly common animal found in the many ponds and water features on campus. We also wanted to use the logo to be light-hearted, encouraging our demographic of students to user the application. The hat, bow-tie and manacle on the duck are an attempt to both add humour and a reference to the clothes swap system.



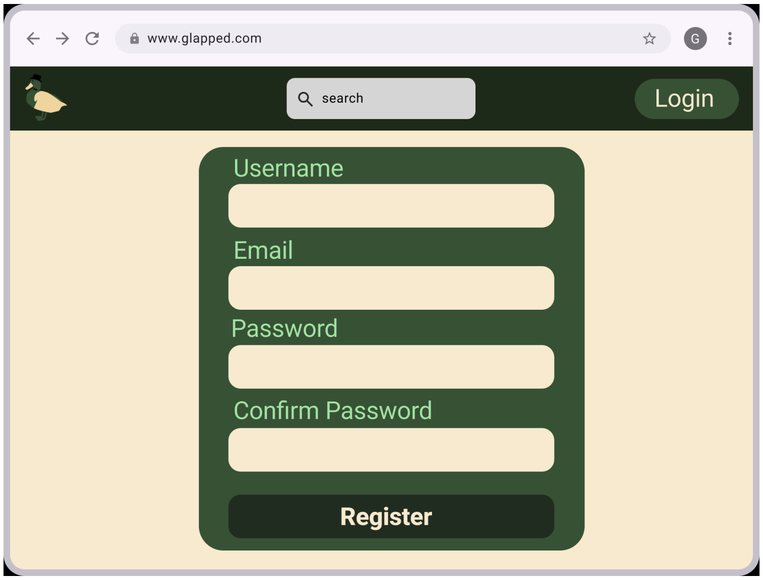
- Icons to encourage gamification - To encourage users to engage with sustainability efforts, how much co2 and water a user has saved by using Glapped will be tracked, and using icons to represent these should help to encourage users to want to engage with their environmental impact in a gamified way.



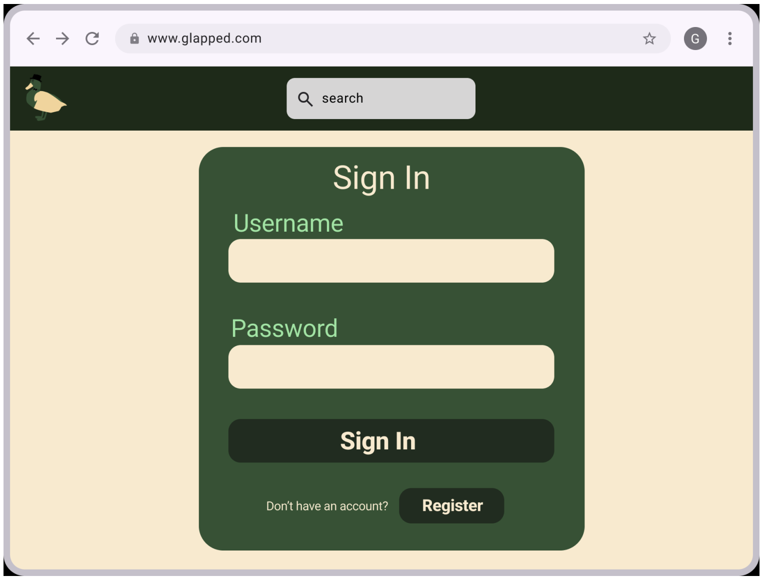
- Landing Page - This is the first page a new user will see, so its design must reflect what will hook users in to use the system. This is achieved by having ”fashion on campus” as the most visible section of text. A new user is then prompted to join, making it easy to get started on Glapped.



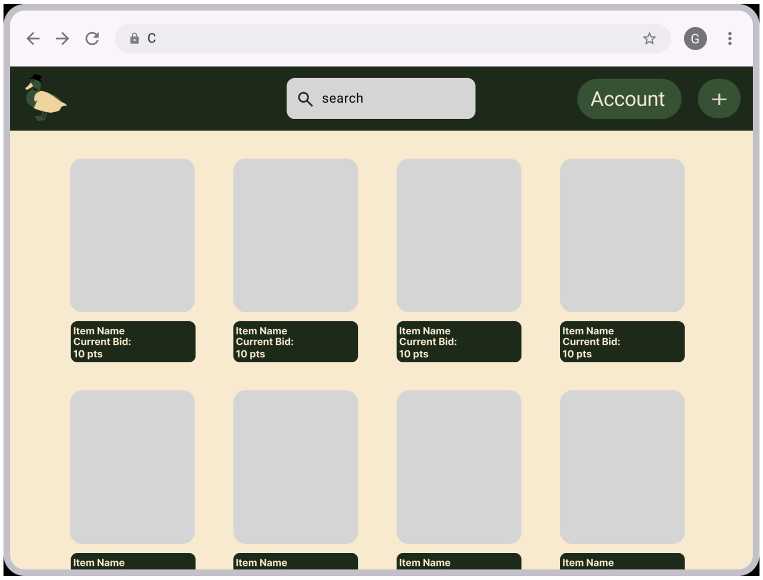
- User Registration Page - A simple user registration page, with minimal extra information to make it as clear as possible and make account creation simple.



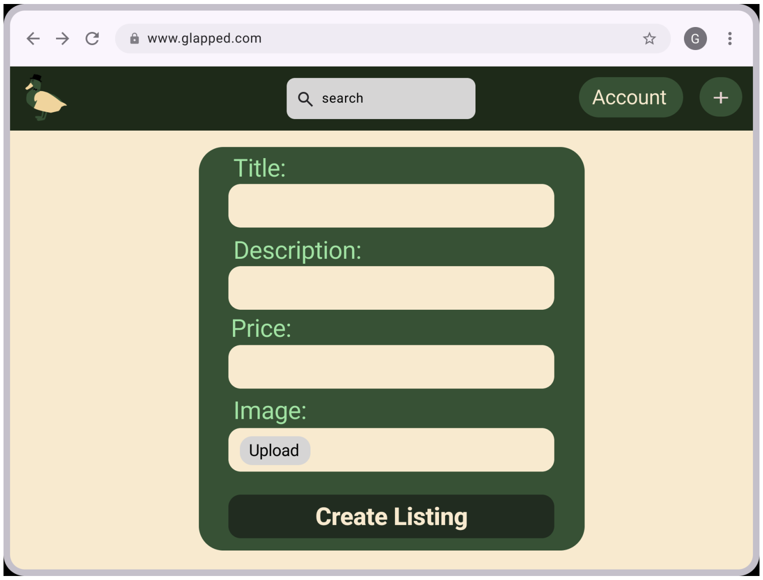
- Sign In Page - A simple sign in page, with a button that directs the user to the register page for the case that a new user mistakes the sign in button with the sign up button.



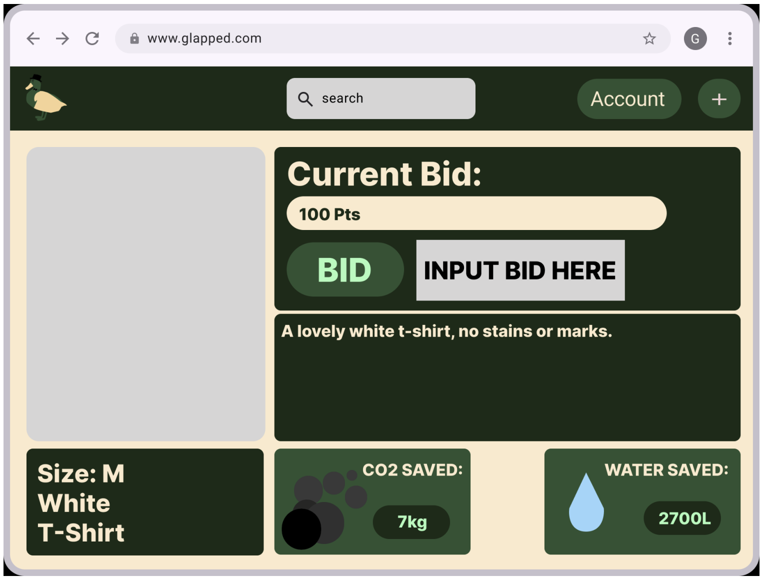
- Home Page/Listings Page - This is the page central to the users experience, this is where they will browse for clothes to bid on. The page will be a scrolling display of every item, with its name and current bid displayed underneath.



- Create Listing Page - Similar layout to the account creation page, with each attribute of a listing needing to be input, and a button to upload images.



- Item/Bidding page - When the user clicks on a listing they will be taken to a bidding page, containing the crucial information on the item, its current bid, the option to bid higher, and the items estimated CO2 and Water saved information.



Current Bid: 100 Pts

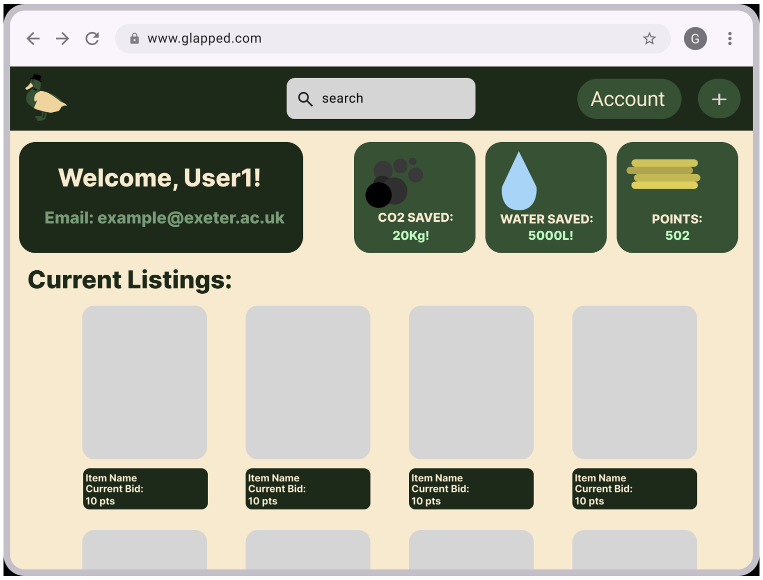
A lovely white t-shirt, no stains or marks.

Size: M
White
T-Shirt

CO2 SAVED: 7kg

WATER SAVED: 2700L

- Account page - This page contains all of the crucial information a user would want to know about themselves, including how much CO2 and Water they've saved and how many points they have, along with any active listings.



Welcome, User1!

Email: example@exeter.ac.uk

CO2 SAVED: 20Kg!

WATER SAVED: 5000L!

POINTS: 502

Current Listings:

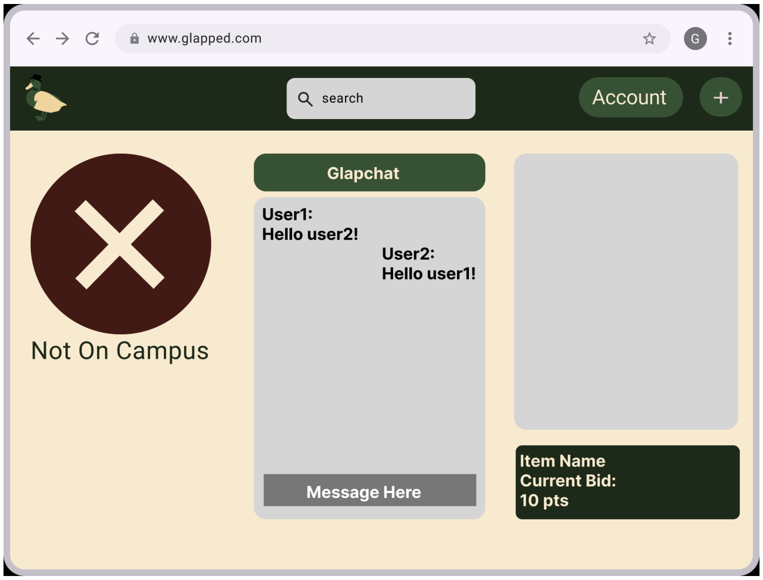
Item Name
Current Bid:
10 pts

Item Name
Current Bid:
10 pts

Item Name
Current Bid:
10 pts

Item Name
Current Bid:
10 pts

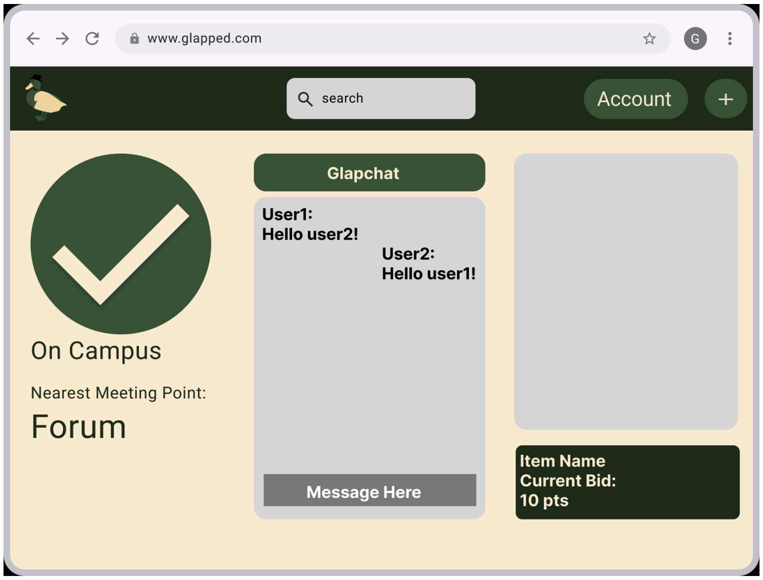
- Communication and Location System - This system will allow two users who are swapping clothes to communicate with each other prior to meeting up, and will alert them when they are both on campus and available to exchange the item. It also contains the basic information on the item in question.



User1:
Hello user2!

User2:
Hello user1!

Item Name
Current Bid:
10 pts

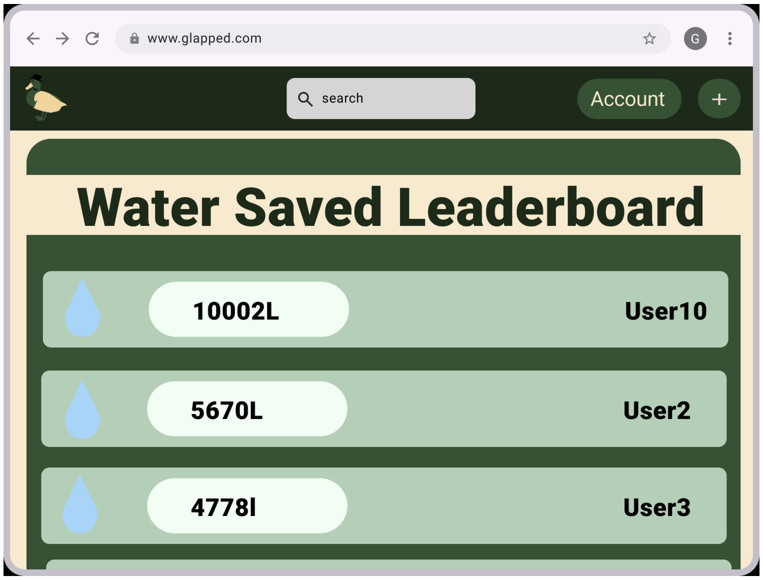


User1:
Hello user2!

User2:
Hello user1!

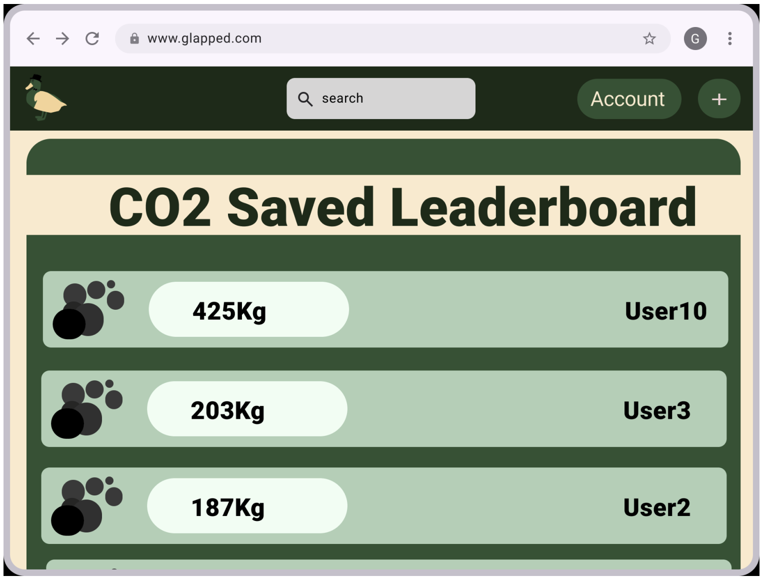
Item Name
Current Bid:
10 pts

- Leaderboards - This system will encourage gamification by rewarding users with placements on a leaderboard to show the positive impact using the application is having on the environment.



Water Saved Leaderboard

10002L	User10
5670L	User2
4778l	User3



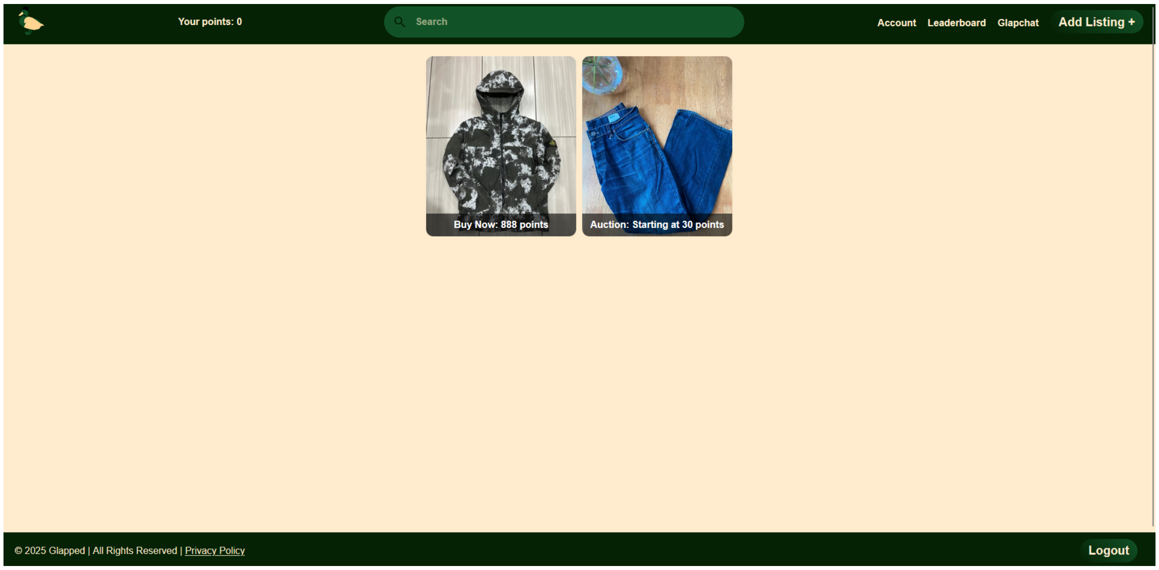
CO2 Saved Leaderboard

425Kg	User10
203Kg	User3
187Kg	User2

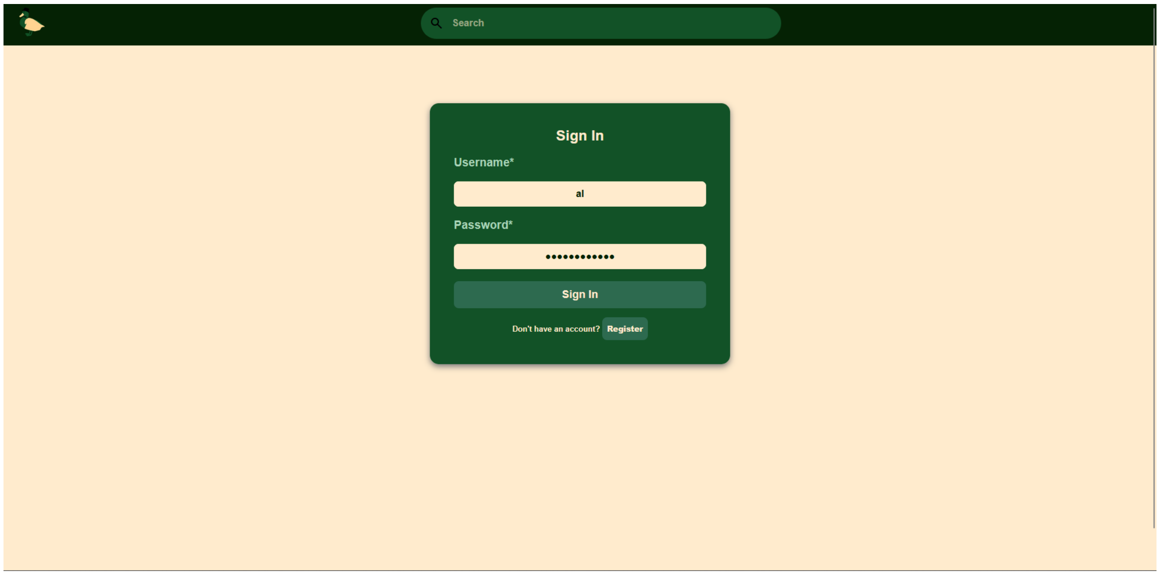
3 The App

The application - Glapped - is based on Django for Python 3.11.4. It has a homepage, a login page, a registration page, a listings page, a page for each specific listing, an account page, and a landing page. In the future we will also have a bidding page, messaging page, a check-in page, and a leader-board page.

- Homepage - The homepage serves as the main place users can check available listings. It shows them images and text details of listings that they can then click through and view. This is achieved by loading the photos from the database, and rendering them in a grid, using a combination of html and css. Each listing then effectively acts as a hyperlink, enabling the user to navigate to the desired listing. There are also animations triggered within the css when the user hovers over an item, to make the site appear more dynamic.



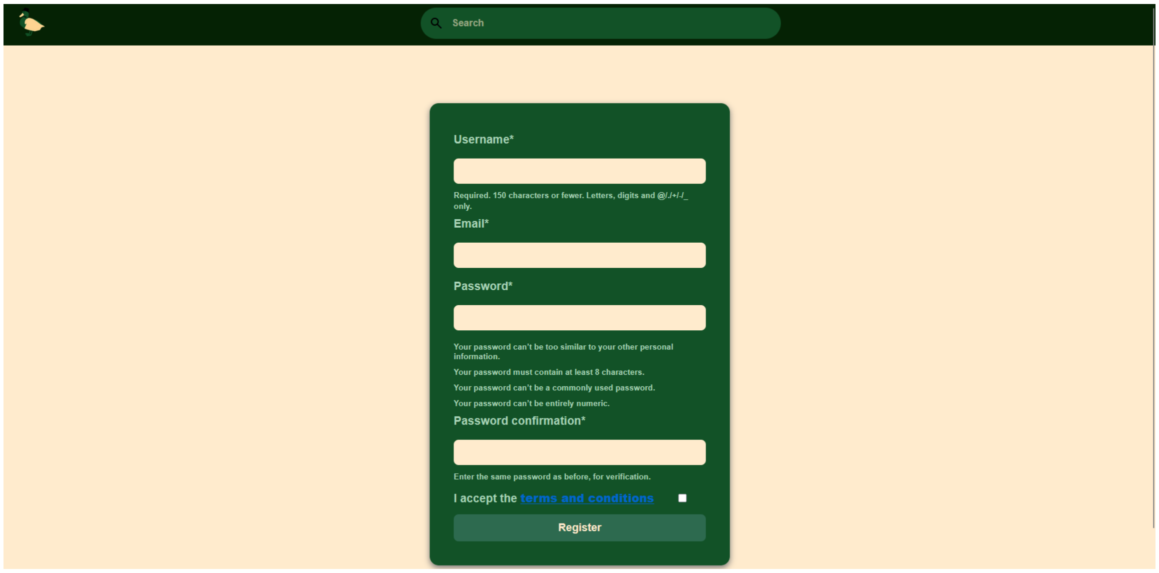
- Login - This uses the Django auth library to handle logins. it can handle and verify login data without storing or passing passwords in plaintext to the back end, instead using a hash. We also implemented a custom css container to make it theme with the website, instead of the bootstrap-style one used by default.



Username*
al

Password*
••••••••••••

- Registration - This also uses the Django auth library. It allows a user to create a profile, which includes their name, username, email, and password. Django can handle making sure passwords are formatted correctly without potential security/GDPR gaps. We also built a custom CSS theme for this form, mirroring the one used for login



Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Password*

Your password can't be too similar to your other personal information.

Your password must contain at least 8 characters.

Your password can't be a commonly used password.

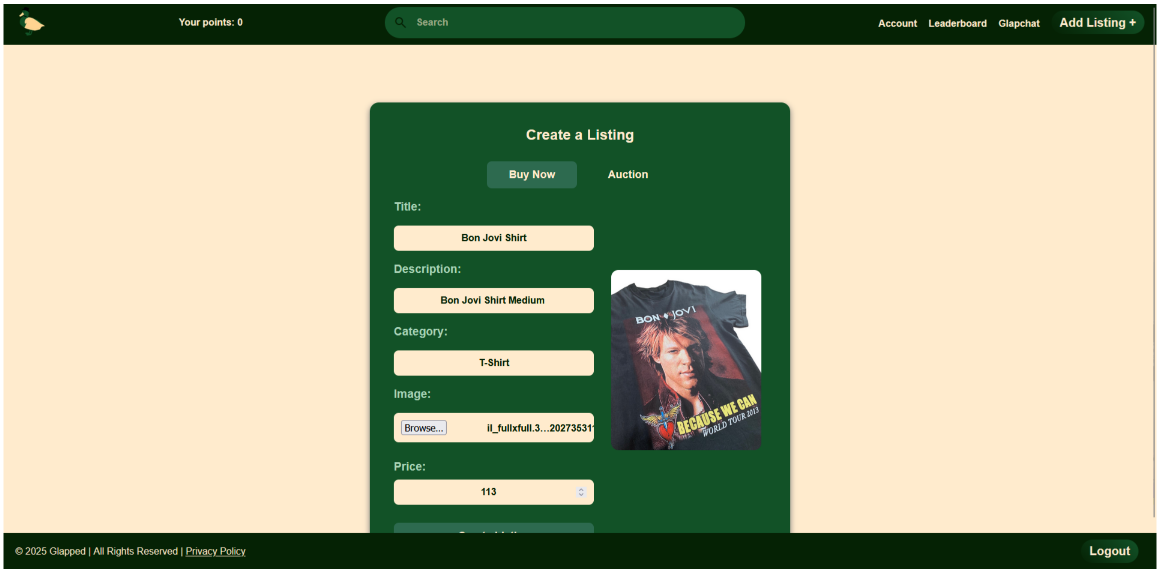
Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

I accept the terms and conditions □

- Create Listing page - This allows users to list items by inputting a form that is then stored in the Django back end database. This form takes a listing name, description, category and an image. You have the option to choose between listing it as an auction, or in a buy it now format. selecting auction will allow you to enter both an auction time, and starting bid. Alternatively buy it now asks only for a price. This also offers a preview of the listing as well as compresses it to prevent people providing files that are too large and slowing the back end.



Create a Listing

Title: Bon Jovi Shirt

Description: Bon Jovi Shirt Medium

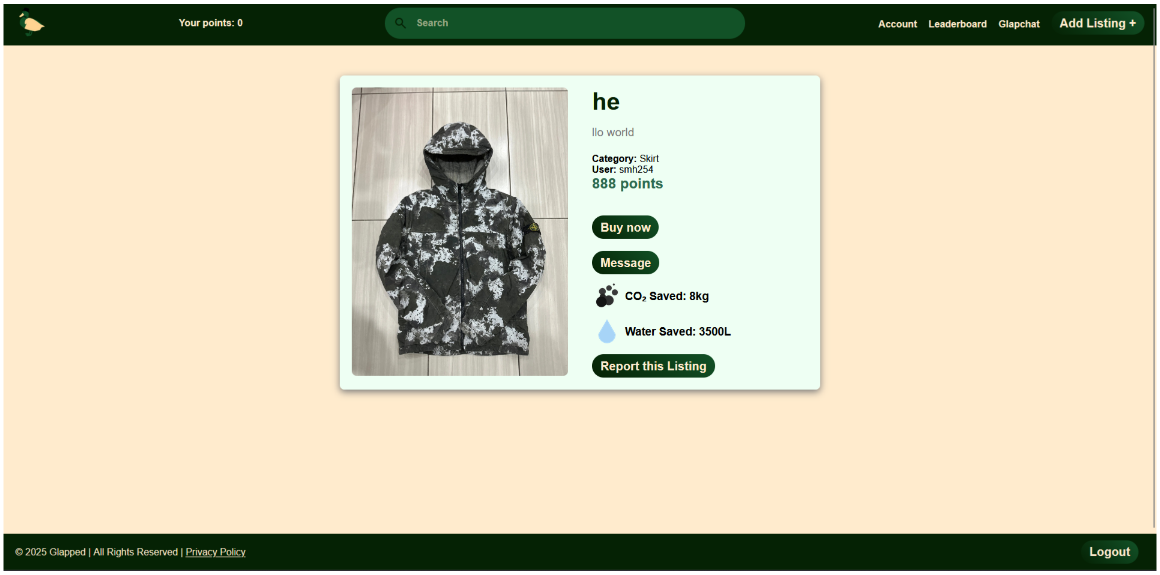
Category: T-Shirt

Image: il_fullxfull.3...20273531

Price: 113

© 2025 Glapped | All Rights Reserved | Privacy Policy

- Listing page - This shows the user the image of the item, as well as its name, a brief description, and a price. It also includes a button for the user to purchase the item. Upon buying an item, the points needed for it will be taken from their account and transferred to the person who listed it. If the selected listing is an auction, it will instead provide the ability to bid on it. You will also have the option to message the user providing the item, via glapchat.



he

llo world

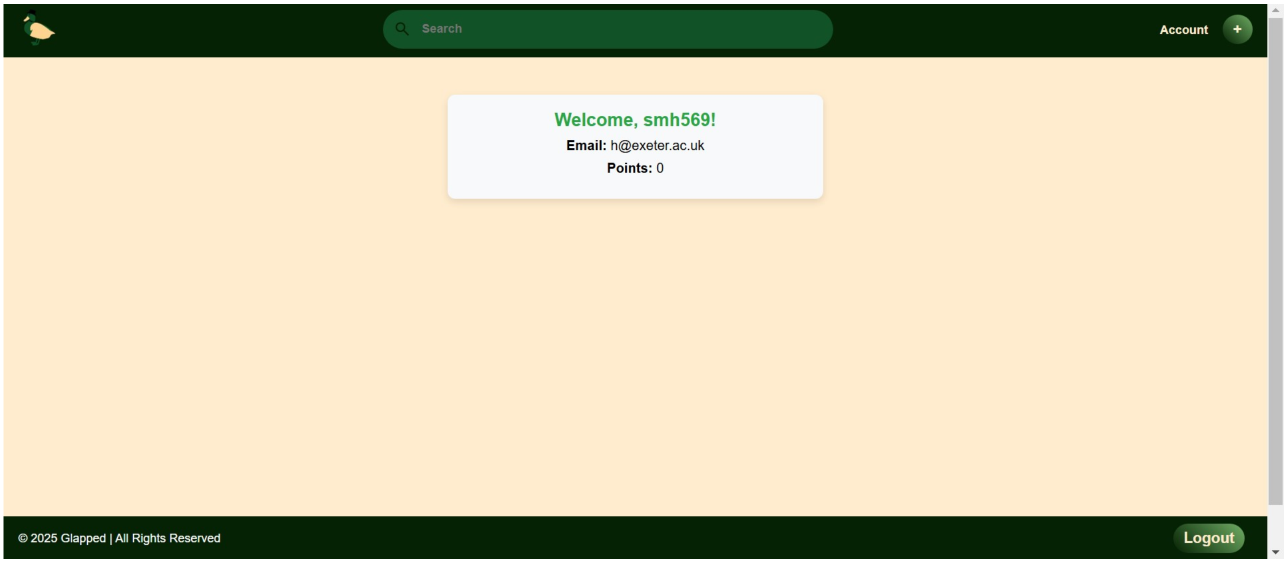
Category: Skirt
User: smh254
888 points

CO$_2$ Saved: 8kg

Water Saved: 3500L

© 2025 Glapped | All Rights Reserved | Privacy Policy

- Account page - This shows the user their username, email, and points count. The points count is updated when a user buys or sells an item. It also shows them their active listed items and their eco-metrics. It hides various details if the user has none, such as hiding the purchased item section when the user has purchased none.



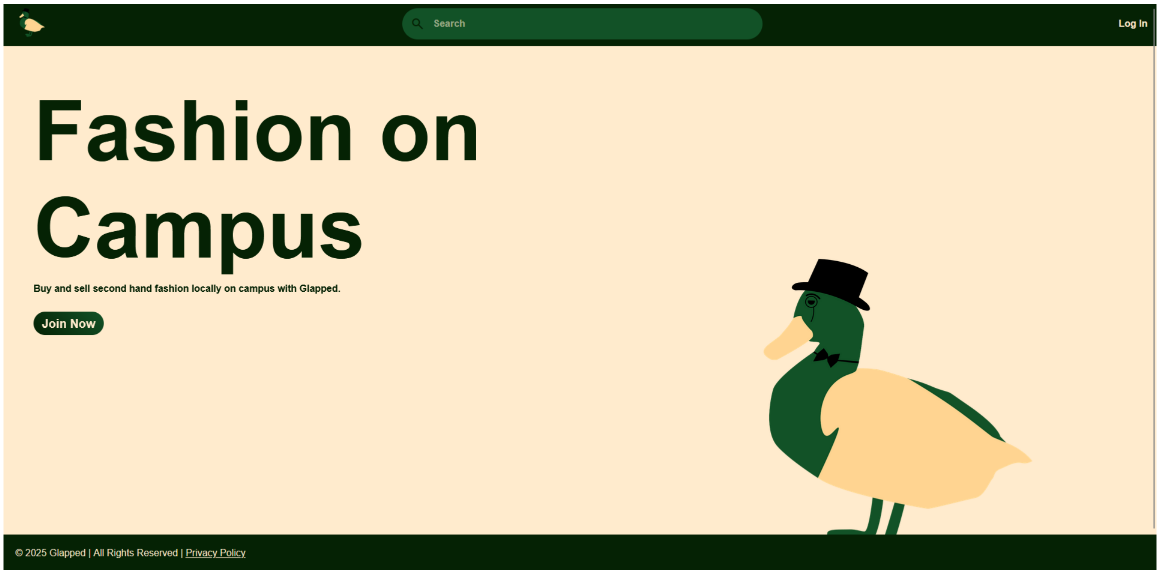
Welcome, smh569!

Email: h@exeter.ac.uk

Points: 0

© 2025 Glapped | All Rights Reserved

- Landing page - This shows the logo and prompts the user to either login or register. It makes use of CSS to perform a slow fade-in animation.

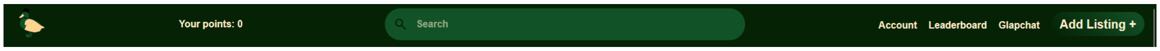


Fashion on Campus

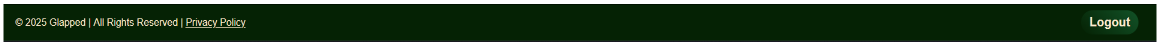
Buy and sell second hand fashion locally on campus with Glapped.

© 2025 Glapped | All Rights Reserved | Privacy Policy

- Navbar - This appears on every page of the website. It varies depending on what section of the website the user is on and weather they are logged in or not. If the user is logged out,it will feature the logo, which doubles as a link to the homepage, and a link to login. when the user is logged in, it no longer shows the user a login option, and gains a search bar, a create listing button, an account button as well as the option to view the leader board. This is done using the html instead of the python back end to dynamically check for when the user is logged in.



- Footer - This has a logout button and a hyperlink to our legal information.



© 2025 Glapped | All Rights Reserved | Privacy Policy

- Leaderboards - Shows a live ranking of the top ten users and their water and CO2 saved. this is done using the python backend to collect the top ten users in each category, before ordering them and arranging them in a grid on the html frontend.

4 Prototype contents

- Homepage
- Login + registration - fully implemented
- Create listing page - fields may be added but otherwise final
- Listing pages - all info displayed, option to buy- no eco-metrics - only buy now, not bidding
- Account page - displays points, current listings/purchases- no eco-metrics - needs more customizability
- Landing page - fully implemented
- Check-in page not implemented, including Glapchat
- Navbar functional
- Footer - legal info not but fully comprehensive otherwise complete

5 Development decisions

App concept was decided on because it allows for a measurable environmental impact and actual use cases for students like:

- safety/convenience - meetups are on campus

- price - items are effectively free, with ”points” used to exchange items
- environmental concerns can easily be addressed through features

Smaller decisions:

- Home page format was chosen due to user familiarity (easier onboarding) and readability
- Meetings on campus verified by location features
- Leader-boards - environmental metrics used to encourage use and eco-consciousness - points leader-board used to encourage users who donate more than they take
- We’ve assumed users will correctly classify their real items for environmental stats - this theoretically enables collusion where two users put up fake items for low cost and repeatedly trade with each other to up their stats.
- Product fields are based on what we assume will be useful

6 Sprint 2 features

- Chat feature
- Environmental stats
- Security
- Location feature for purchased items meetup
- Option to bid on items instead of ”buy now”
- Better user profiles
- Issue - image size adjustments
- Issue - system exploits (mass account creation, wash trading)
- Way to gain small amount of points daily
- More test coverage
- GDPR documentation + other legal stuff
- Issue - moderation is needed for public use

7 Updating Glapped

The overall team was split into a number of subgroups to handle the tasks outlined for the future sprint features. The first things that were tackled were the bidding systems, chat features, and moderation. These were done first as they were deemed to be the minimum parts required to realise an environment that can self regulate value. After each was implemented, the next item on the checklist was implemented according to value added. As well as this, though, all the issues were systematically removed.

8 Chatting

The chat feature allows for users to be connected through the website for sending offers / booking times to meet and swap items. The chat is either accessed from a button on the item listing which will create a new chat or from a button in the page’s navbar. The chat page has a dialogue box for users to input their messages as well as a display for all messages in the chat. 2 users chat in a specific “room”. this room has an ID and a page where sending a text will create a secure websocket between the users and creates a consumer object. This is routed with routing.py and goes through an nginx server. thanks to a daphne protocol server. The consumer deals with how the websocket works. one of the things it does is when a websocket is “received” it creates a new attribute for the message entity in the database, and updates the screen for other users without reloading the webpage.

9 Moderation

Moderation works by the user base reporting content that they may find disagreeable before the moderators manage it. When a report is made, a user will press the report button which then links them to a page that has a form to detail their report. A user can then fill in the form provided they haven't tried to report the same item before. Once a listing has more than 3 reports, it will no longer be shown to users automatically. Admins can review the reports and decide if they should be used and either remove the listing forever or continue to show it to users.

10 Bidding

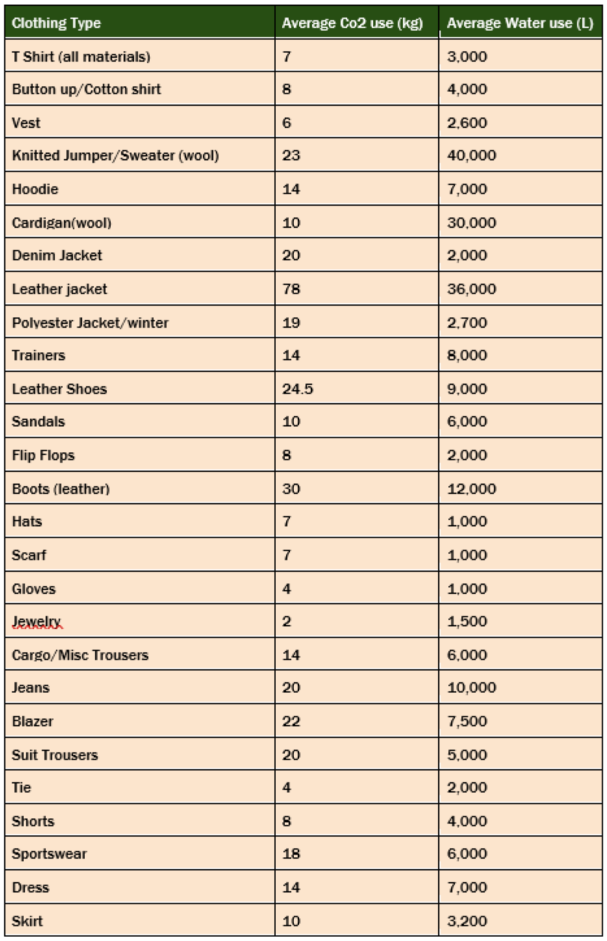
Upon creating a listing a user can choose to either have it be a fixed price listing or an auction. When a user selects the Auction option, they can then specify details about the item as well as a starting price and an auction length. This is then shown on the homepage as specifically an auction. A user can then click on this item and choose to bid higher by inputting a bid. If they are successful in placing a bid, it will temporarily deduct points from their account before updating who is the current winner. This is to prevent a user from winning a bid after buying an item that puts them in debt. If someone else rebids the last winners points are returned to them. Once a bid is won, the points are taken from the winners account and they can claim the item. Bidding items and Buy now items both inherit from a generic product class to be stored in the database.

11 Data Policy

Data policy was written to fall in line with UK government advice. These meant that we implemented a clarification section to the registration page, as well as a number of methods for users to remove their information from the system.

12 Sustainability

We have obtained estimates of the amount co2 and water required to produce different categories of clothing. These were sourced from a number of places, and where data couldn't be found, extrapolated from material cost and weight.



Clothing Type	Average Co2 use (kg)	Average Water use (L)
T Shirt (all materials)	7	3,000
Button up/Cotton shirt	8	4,000
Vest	6	2,600
Knitted Jumper/Sweater (wool)	23	40,000
Hoodie	14	7,000
Cardigan(wool)	10	30,000
Denim Jacket	20	2,000
Leather jacket	78	36,000
Polyester Jacket/winter	19	2,700
Trainers	14	8,000
Leather Shoes	24.5	9,000
Sandals	10	6,000
Flip Flops	8	2,000
Boots (leather)	30	12,000
Hats	7	1,000
Scarf	7	1,000
Gloves	4	1,000
Jewelry	2	1,500
Cargo/Misc Trousers	14	6,000
Jeans	20	10,000
Blazer	22	7,500
Suit Trousers	20	5,000
Tie	4	2,000
Shorts	8	4,000
Sportswear	18	6,000
Dress	14	7,000
Skirt	10	3,200

13 Security Considerations

Security measures include:

- Authentication and Authorization: Django's built-in authentication system handles user login and registration, hashing passwords securely. Role-based access control (RBAC) ensures restricted access to administrative functions.
- Data Protection: User passwords are encrypted using Django's default hashing algorithm.
- Form Validation: Input fields are sanitized to prevent SQL injection and XSS attacks.

14 Hosting

Hosting the site is provided by amazon web services and runs at the domain glapped.store. Setting this up required amazons web services as well as an elastic ip. This host is responsible for running daphne powered django channels which are used for NGINX as a webserver and reverse proxy. As well as this the server also sends and recieves standard https protocol information.

15 Testing

The majority of the testing is provided by unit tests. Testing was done as we went along, mostly following a test-driven development style. As we were programming in smaller subgroups we could have members split duties between test writing and production code.

16 Gamification

Glapped aims for a Gamified environment by encouraging sustainable behaviour through user competition. This is done via leader boards, which incentivise increased partition through public recognition.

Gamification is also aided via the introduction of a points economy. Rewards incentivise users returning, while the fact that points can only be spent on other listings introduces a circular economy where we directly reward sustainability.

we also use icons and metrics in the account section, to provide direct and immediate visual feedback for users to encourage activity on the platform.