

Core Backend Features

1. Authentication & User Management

- Google Login (OAuth2)
- JWT-based auth
- Route protection (only allow logged-in users to create/update/delete)
- User profile (basic info + preferences)

API Endpoints:

- `POST /auth/google-login`
- `GET /auth/me`
- `POST /auth/logout`
- `PUT /user/profile`

2. Product Management

- Add a new product (with Cloudinary image upload)
- Edit product
- Delete product
- View individual product
- Filter by tag/category
- Filter by location
- Mark as sold

API Endpoints:

- `POST /products` – Add product
- `GET /products/:id` – View product
- `GET /products` – Home feed
- `PUT /products/:id` – Edit product
- `DELETE /products/:id` – Delete product
- `PATCH /products/:id/sold` – Mark sold

Properties of a product:

Field	Type	Notes
<code>id</code>	string	UUID or MongoDB ObjectId
<code>name</code>	string	Product name
<code>description</code>	string	Optional
<code>price</code>	number	
<code>date_added</code>	datetime	Auto
<code>seller_id</code>	reference to user	
<code>image_urls</code>	array of strings	Stored in Cloudinary
<code>location</code>	string	eg. "Hostel C"
<code>category</code>	string	eg. books, electronics
<code>tags</code>	array of strings	For search
<code>is_sold</code>	boolean	
<code>views</code>	number	Optional
<code>likes</code>	number	Optional
<code>fuzzy_embedding</code>	vector	For semantic search (if stored)

3. Fuzzy Search (Semantic Search)

- Use `sentence-transformers` to convert product name/description to embeddings

- Store vector in DB (MongoDB via Beanie + vector plugin, or use an external vector DB like Pinecone)
- On search: convert query → embedding → return top-k matches

API Endpoint:

- `POST /search` – takes a query string and returns top matches

Note: If not storing embeddings in MongoDB (with `beanie-vector` or similar), you might need an in-memory vector store + periodic refresh.

4. Home Feed

- Latest products
- Filter by category
- Option to prioritize items near user's hostel (if location given)

API Endpoint:

- `GET /products?sort=newest&location=Hostel%20C&category=electronics`

6. Likes / Saved Items

- Users can like/save products
- View saved items later

API Endpoints:

- `POST /products/:id/like`
- `GET /user/saved`

7. Report a product

- Report a product
- Admin dashboard (moderate reported content)

API Endpoints:

- `POST /products/:id/report`
- `GET /admin/reports`

8. Image Handling

- Upload to Cloudinary via signed URL or server-side middleware
- Compress image before upload (frontend or backend)
- Delete image on product deletion

Flow:

1. Frontend → selects image
2. Backend → uploads to Cloudinary
3. Backend → saves URL in product record

Interaction between buyer and seller

API Flow

Buyer

- Sees anonymous product listings.
- Clicks “**I’m Interested**” → interest record created.
- Gets confirmation: “Your interest has been sent to the seller.”

Seller

- In profile/dashboard:
 - Sees all interested users for each product.
 - Each user is shown with: email/phone, and optional message.
 - Seller can:
 - Click “Reveal contact” (if obfuscated initially)
 - Click “Reply via WhatsApp/email”
 - Ignore or block user