

Table of contents:

[Option 8: Buy/Sell Listing Website](#)

[User stories](#)

[ERD](#)

[Wireframes](#)

[Routes / page setup](#)

[EJS Views](#)

[EJS Partial](#)

[Tech choices](#)

[Possible stretch features](#)

[Working log](#)

Option 8: Buy/Sell Listing Website

An app where you can put different types of things up for sale. You can pick a specific niche of items to sell for the app (a cars site, a shoes site, etc). This lets buyers find the items they are looking for quickly, and easily contact sellers.

User requirements	Admins can:
<ul style="list-style-type: none">• users can see featured items on a main feed• users can filter items by price,• users can favourite items to check up on them later• users can send messages to the user that is listing the item	<ul style="list-style-type: none">• post items, which can be seen by others• remove items from the site• mark items as SOLD!,• send a message via app, email, or text back on negotiations in buying the said item

User stories

- As a user, I want to see featured cars on my main feed, because it's easier than searching and I am interested in what is popular

- As a user, I want to filter cars by price/specific details, because I have a budget or might be looking for a specific car
- As a user, I want to favourite/save a car
- As a user, I want to communicate with the lister to ask questions or buy their car
- As a user (admin), I want to be able to list a car, because I want to sell a car
- As a user (admin), I want to be able to remove a listing, because I've changed my mind
- As a user (admin), I want to mark a listing as sold, because it is no longer available
- As a user (admin), I want to respond to any messages/questions

ERD

https://lucid.app/lucidchart/2c5f12b0-048b-48ac-8291-8acb795ef7d1/edit?viewport_loc=-56%2C-40%2C1749%2C968%2COcff1lWeMpUK&invitationId=inv_9b892b7d-c4b5-4c92-9015-6e4cc87e2a02

Wireframes

https://docs.google.com/drawings/d/1rpnVJGTuE84UdVcv6FAfbxNiggEuztCf_LlgF3rTyBc/edit

- Home - featured listings
- Search -
- My favourites -
- My listings - admin stuff
- Messages -
- Login/Register - *won't actually work?

Routes / page setup

ROUTERS	PATH	GET	POST	PUT	DELETE
main/server	/	Display homepage	N/A	N/A	N/A
favourites	/favourites	Display user favs	Add to user favs	N/A	Set is_deleted to TRUE
listings	/listings	Display 'All listings'	N/A (could move POST /listings/new to here)	N/A	N/A
	/listings/new	Display new listing form	Post new listing	N/A	N/A
	/listings/:id	Display single car listing, where user can send message/favourite	N/A	Owner can update listing as sold	Set is_deleted to TRUE

messages	/messages	Display user inbox	N/A	N/A	N/A
	/messages/:id	Display convo	Send message/reply	N/A	N/A (stretch?)
main/server	/login	Login page	Login user/set cookie	N/A	N/A
	/login/:id	Login directly to userId	N/A	N/A	N/A
	/register	Register page	Add user to DB and login/set cookie	N/A	N/A
	/logout	N/A	Logout user	N/A	N/A
users-api	/api/users	Displays JSON of all users			
listings-api	/api/listings	Displays JSON of all listings			
favourites	/api/favourites	Displays JSON of all favourites			
messages-api	/api/messages	Displays JSON of all messages			

EJS Views

index - homepage, featured

favourites - user favs page

all-listings - all active listings with search option (default load with all active, then use AJAX for showing filtered results?)

listings - my listings

listing-show - page to display single car listing

listing-new - post new listing form

messages - user inbox

messages-show - single convo

login - login page

register - register page

error - error splash

email / phone - forms for sharing listings

EJS Partials

_head - <head> content

_nav - navbar

_footer - things to be loaded at page bottom (actual footer or any deferred scripts)

Tech choices

- Back End: Node & Express
- Front End: HTML (ejs) , CSS, JS, jQuery, Bootstrap

Possible stretch features

- Deleting messages
- Hits count/stats for listing pages

Working log

Date	Stuff done
Dec 17 (TAI)	<p>branch: listings-setup >>> <i>MERGED INTO</i> >>> master</p> <ul style="list-style-type: none">• /db/queries/listings.js - getListings, getListing, getUserListings, getMostPopular• /db/schema - updated cars table to include timestamp to be the posting date• /db/seeds - updated all seeds to include more data, comments left at the bottom of the files to show relations between the data for debugging/testing (e.g. making sure proper messages show up in inbox once we get there)• /public/scripts/index.js - AJAX request added to fetch featured cars from the /api/listings/featured route• /public/scripts/listings.js - AJAX requests to fetch all listings or user-specific listings based on how the listings EJS page was rendered (either as 'All Listings' or 'My Listings' based on the initial GET/ path)• /routes/listings-api.js - API routes where database data is displayed as JSON• /routes/listings.js - regular routes for EJS rendering (the EJS templates then pull data from the /api/ routes)

	<ul style="list-style-type: none"> • /styles/main.scss - changed font to something that looks more premium, fixed misc spacing stuff and made nav text all caps, also added some formatting to ensure any EJS templates that load, don't get stuck behind the nav • /views/partials/_nav.ejs - made the logo disappear when width ~1200px to prevent nav options from being obscured • /views/index.ejs - commented out the three cars that you added last night as I moved them to render via <code>/public/scripts/index.js</code> - make any changes to the HTML template there • /views/listings.ejs - page title & H1 will either say 'All' or 'My' depending on a variable passed through by <code>/routes/listings.js</code> • /server.js - added <code>/login/:id</code> and <code>/logout</code> routes so we can login directly to a userId for testing. user id 3 has 3 car listings if you want to see their My Listings page. <p>Notes:</p> <ul style="list-style-type: none"> • The skeleton code they gave us was broken down into the <code>/api/</code> routes to hold any DB queries as a JSON object. If we want to use AJAX to dynamically load some stuff to pages (reduce the number of EJS templates we need) then I think keeping it this way might be best, but there's definitely opportunity to change how everything talks to eachother
Dec 18 (Edward)	/views/index.ejs Added search box to the main page.
Dec 19 (Tai)	<ul style="list-style-type: none"> • /views/index.ejs - updated search box to overlay the car image. controls for how it overlays is in main.scss under <code>#search-container</code> • /db/queries/users.js - set up basic functions for db connection, <code>getUserBy[Id, Email, Name]</code> and <code>addUser</code> (none are functional / in use by any pages) • /routes/listings.js - fixed bug where logged out user could not access All Listings/My Listings • /server.js <ul style="list-style-type: none"> ○ Added an <code>authorizeUser</code> function to be run before some pages load. This currently just checks if the user has a <code>user_id</code> cookie and renders the <code>error_template</code> if not. ○ Updated all existing routes (visible to a logged in user) to provide a <code>templateVars</code> with their <code>userId</code>, to ensure the <code>_nav.ejs</code> partial can switch between Login/Register or Logout accordingly ○ Added catch-all for urls that don't exist (<code>GET /*</code>) to render a 404 page • /views/error_template.ejs - created error template that can be rendered as needed. It takes an <code>errorVars</code> object with the following keys: <ul style="list-style-type: none"> ○ <code>status</code>: http status code ○ <code>message</code>: error message to be displayed

	<ul style="list-style-type: none"> ○ cta: optional call to action, which is an object that has a url and display key, to render a clickable link below the error status/message ● /views/login.ejs - created login landing page (copied from my TinyApp) - NOT FUNCTIONAL ● /views/register.ejs - created register landing page (copied from my TinyApp) - NOT FUNCTIONAL ● /views/partials/_nav.ejs - added IF statement to show logout instead of login/register if a userId was sent in the templateVars for the page ● /views/listings.ejs - Made listings cards entire body clickable, also added fixed height (update in main.scss if/when needed: .card-img-top and .card - will have to re-test with longer car descriptions) ● Added custom logo to replace the one borrowed from real site
Dec 19 (Edward)	<ul style="list-style-type: none"> ● /server.js Add favoritesApiRoutes and favouritesRoutes ● /views/favourites.ejs copied template to file ● /routes/favourites-api.js API routes where database data is displayed as JSON ● /routes/favourites.js regular routes for EJS rendering (the EJS templates then pull data from the /api/ routes) ● /public/scripts/favourites.js Ajax request to get user favourites ● /db/queries/favourites.js Selects the favourite list for a certain user along with the cars information ● I did a merge and resolved a conflict on github.com but doesn't look like anything deleted. ● Found out after I was supposed to remove the older header but found out after there were 2 headers merged so I deleted the old one and left the new one. ● /server.js Add searchApiRoutes and searchRoutes ● /views/search.ejs Add file with search by body types copied from auto trader ● /routes/search-api.js I need to add code for this (Not working at the moment) ● /routes/search.js added the get / route ● /public/scripts/search.js Need proper code (Not working at the moment will update) ● /db/queries/search.js Need to add proper queries
Dec. 21 (Edward)	<ul style="list-style-type: none"> ● /db/queries/search.js Added test search query(Need to update so that it uses the form data) ● /public/scripts/search.js Added Ajax request ● /routes/search-api.js Add to show json data for search ● /routes/search.js Add post request for the make, model, min year and max year form (but need to update so that I can use the data from the form in the search query) ● Github I did a push with the search branch and then I merged it.
Dec. 23	<ul style="list-style-type: none"> ● /db/queries/search.js Added search queries for make, model, min year, max year and body type

(Edward)	<ul style="list-style-type: none"> • /views/search.ejs Added car makes and body type search href • /db/schema Added body_type to cars table • /db/seeds Added body_type to car values
Dec. 25 (Edward)	<ul style="list-style-type: none"> • /views/index.ejs Added main image slideshow • /styles/main.scss Added styles for the main image slideshow
Dec. 27 (Edward)	<ul style="list-style-type: none"> • /views/search.ejs Added min and max price • /routes/search-api.js Add to get min price and max price values • /db/queries/search.js Update query to use min price and max price values • /styles/main.scss Removed duplicate code • /views/partials/_footer.ejs Added a footer partial • /styles/footer.scss Added footer styles • /public/styles/footer.css Added file • /views/partials/_head.ejs Added 2 footer stylesheets • Added the _footer link to the index.ejs, search.ejs, listings.ejs, favourites.ejs, register.ejs and login.ejs
Dec. 28 (Edward)	<ul style="list-style-type: none"> • /views/search.ejs Added green body type images for on hover event • /styles/main.scss Added to change to green car image when hovered over body type
Dec. 29 (Edward)	<ul style="list-style-type: none"> • /views/search.ejs Added min and max kilometer and sort by, added image • /views/listings.ejs Added image
Dec. 30 (Edward)	<ul style="list-style-type: none"> • /db/queries/search.js Add query to order by default is by make • /routes/search-api.js Add to get sort by selection • /db/schema Added kms to cars table • /db/seeds Added kms to car values
Jan. 1st (Edward)	<ul style="list-style-type: none"> • /db/seeds Updated some seed values • /views/search.ejs Updated look • /views/listings.ejs Updated the styling and formats
Jan 2 (Tai)	<ul style="list-style-type: none"> • Minor fixes to nav styling/prevent content from page from peeking through on the sides

	<ul style="list-style-type: none"> • Fixed some of the vehicle dropdowns (fixed value of 2023 in dropdown, added 2024 & 2025, fixed duplicate Hyundai option and added more vehicle makes) • /db/queries/listings.js Add query to insert new listing into DB • /public/scripts/new-post.js Add jquery listeners to update mileage/price based on input sliders, and ajax handling of form submission • /routes/listings.js Add POST handling for /listings/new • /views/listing-new.ejs Created form for adding a new listing • /public/scripts/favourites.js & /views/favourites.ejs Updated favourites.ejs & favourites.js to render user favs into rows, similar to search & listings views
Jan. 2nd (Edward)	<ul style="list-style-type: none"> • /views/search.ejs Updated timestamp to remove seconds • /views/listings.ejs Updated timestamp to remove seconds • /views/listing-show.ejs Added file but can't get it to retrieve the data from /api/listings/5 for example • /public/scripts/listing-show.js Added code to try to get /api/listings/4 for testing but doesn't work
Jan 3 (Tai)	<ul style="list-style-type: none"> • /queries/messages.js Created DB query for messages (single function to either return all messages for a userId (limited to most recent, per listing convo) or all messages for a specific car) • /public/scripts/messages-inbox.js Ajax loading messages to the /messages inbox, sorting into three different areas: Outgoing messages (from user to a buyer), Incoming messages (from a user to YOU, as the seller), or Archived (for anything where sold = true) • /routes/messages-api.js API routes for messages, will either return latest message in each thread (GET /api/messages/user/:userid) OR all messages between a car & the buyer (GET /api/messages/car=:carid&buyer=:buyerid) • /routes/messages.js Regular routes for loading message inbox or specific convo (GET /messages will load the inbox page, GET /messages/:carid/:buyerid will load a convo) • /public/views/messages.ejs Inbox page created, rendering successfully, href links to correct destination but messages-show has to be created still
Jan. 3rd (Edward)	<ul style="list-style-type: none"> • /views/listing-show.ejs Removed html template that shows car listing and put div with id listing-show • /public/scripts/listing-show.js Added the html template and variables • /db/schema added 05_car_details table • /db/seeds added 05_car_details values • /queries/listings.js Updated carListing query to Left join the car_details table • /styles/main.scss Add styling for listing-show.ejs

<p>Jan 4 (Tai)</p>	<ul style="list-style-type: none"> • /views/listing-new.ejs Updated with additional fields for car details (included drivetrain dropdown if you want to add it to the schema) • /public/scripts/new-post.js Added script for fuel consumption input slider, it will display/submit as N/A if set to zero • /db/queries/listings.js Updated to perform second insert into car_details using the car_id returned from initial DB request. Confirmed both databases get updated and listing-show displays all of the correct info that was submitted. 🥳 • /db/queries/messages.js <ul style="list-style-type: none"> ◦ Fixed getMessages function to properly return partitioned results where user=buyer and user=lister ◦ Added addMessage function to add message into DB, taking 5 params: (userId, carId, listerId, buyerId, message) – userId is compared to the listerId to determine whether the message entry is a reply or not (how DB is structured to determine if a reply is to or from a lister) • /seeds/cars.sql Fixed car 10 to be owned by user 8 as it was changed in error • /seeds/messages.sql Replaced placeholder messages with fake conversations to ensure they are threading correctly in the messages-show rendering • /public/scripts/messages-convo.js load car info & threaded messages on page load, and handle form submission to POST /messages • /public/scripts/messages-inbox.js minor changes, now shows 'Last message from you' or 'Last message from USERNAME' as a preview for the message • /routes/messages-api.js Temporarily removed userId check to ensure listers could access their messages (will re-implement a check later) • /routes/messages.js POST routing added for new messages. Can leverage for the Listing pages when sending initial message • /views/messages-show.ejs Threaded convo view set up • /public/scripts/listing-show.js Modified listing-show rendering to only show the Contact Seller form to logged in users <ul style="list-style-type: none"> ◦ If user is not logged in, it will ask them to login/register ◦ If the user owns the listing, it has placeholder text, where we can load a button to mark a listing as sold or deleted • /views/listing-show.ejs Added hidden span at bottom of page for the userId & listerId (used by the form to send messages)
<p>Jan 4th. (Edward)</p>	<ul style="list-style-type: none"> • /public/scripts/listing-show.js Add edit and delete buttons, update to that thumbnail images are clickable and display the larger image

	<ul style="list-style-type: none"> ● /public/scripts/index.js Add favorites heart ● Need to update favorites button to run addUserFavorite() and get the current logged in user and car id ● /seeds/cars.sql Updated and added some real data ● /seeds/car_details.sql Added additional information so that car picture will show
Jan 5 (Tai)	<p>Marking as sold/deleted, ensuring deleted listings do not show in other queries</p> <ul style="list-style-type: none"> ● /db/queries/listings.js <ul style="list-style-type: none"> ○ Added queries for setting listings as sold/unsold, delete/undelete ○ Updated getAll to sort by newest first with sold/deleted last ● /public/scripts/listings.js Updated so it doesn't show deleted listings and sold listings have a badge ● /public/scripts/listings-show.js <ul style="list-style-type: none"> ○ Minor tweak to send to phone/email, add to fav styling ○ Added onClick event for addFav and removeFav ○ Remove from favourites will be set after page loads, second ajax call to /api/favourites/user/:id to determine if the car is favoured ○ Updated to show SOLD beside the price for a sold listing ○ Updated sidebar to show 'Active Listing, Sold Listing, or Inactive Listing' and posted timestamp from cars.timestamp ○ Updated sidebar for OWNER: <ul style="list-style-type: none"> ■ Edit listing button (only shows for available & active car – non functional at this time) <ul style="list-style-type: none"> ● Not sure if we have time to implement actual edit functionality ■ Mark as sold / as available: implemented & sets cars.sold to TRUE or FALSE ■ Mark inactive / active: implemented & sets cars.is_deleted to TRUE or FALSE ● /public/scripts/favourites.js Updated so Favourites list shows a badge if a listing is sold, and to not list 'deleted' listings ● /routes/listings.js PUT routing for /listing/:id <ul style="list-style-type: none"> ○ Receives an object containing four possible key/value pairs <ul style="list-style-type: none"> ■ sold: 'true' -> runs markAsSold query ■ sold: 'false' -> runs markAsUnsold query ■ deleted: 'true' -> runs markAsDeleted query ■ deleted: 'false' -> runs markAsUndeleted query <p>Favourites functionality:</p> <ul style="list-style-type: none"> ● /db/queries/favourites.js Added queries for adding and removing user favourites

	<ul style="list-style-type: none"> • /db/schema/04_user-favs.sql Removed is_deleted boolean from user favs, easier to just remove record entirely when user un-favs a listing • /routes/favourites.js POST and DELETE routing <ul style="list-style-type: none"> ◦ POST /favourites runs the addUserFavourite ◦ DELETE /favourites runs the removeUserFavourite • /views/partials/_nav.ejs Updated favourites to just be /favourites instead of /favourites/mine
Jan. 5th (Edward)	<ul style="list-style-type: none"> • /routes/search.js Added to send search options chosen to search.ejs, if not chosen it will send default values • /views/search.ejs Add to show what search options where chosen, also add message if no results found
Jan 6 (Tai)	<p>New logo</p> <ul style="list-style-type: none"> • Added new logo/company name, replaced 'Call us' in header to a link to a fake FAQ page <p>Edit listing price:</p> <ul style="list-style-type: none"> • /db/queries/listings.js Add updatePrice function • /public/scripts/listing-show.js Add editPrice logi • /routes/listings.js Add PUT routing for {price: } obj <p>Add/remove favs via Index, Search & All Listings (heart icon is hidden for guests)</p> <ul style="list-style-type: none"> • /public/scripts/listings.js Added ability for logged in users to add/remove favs from other pages