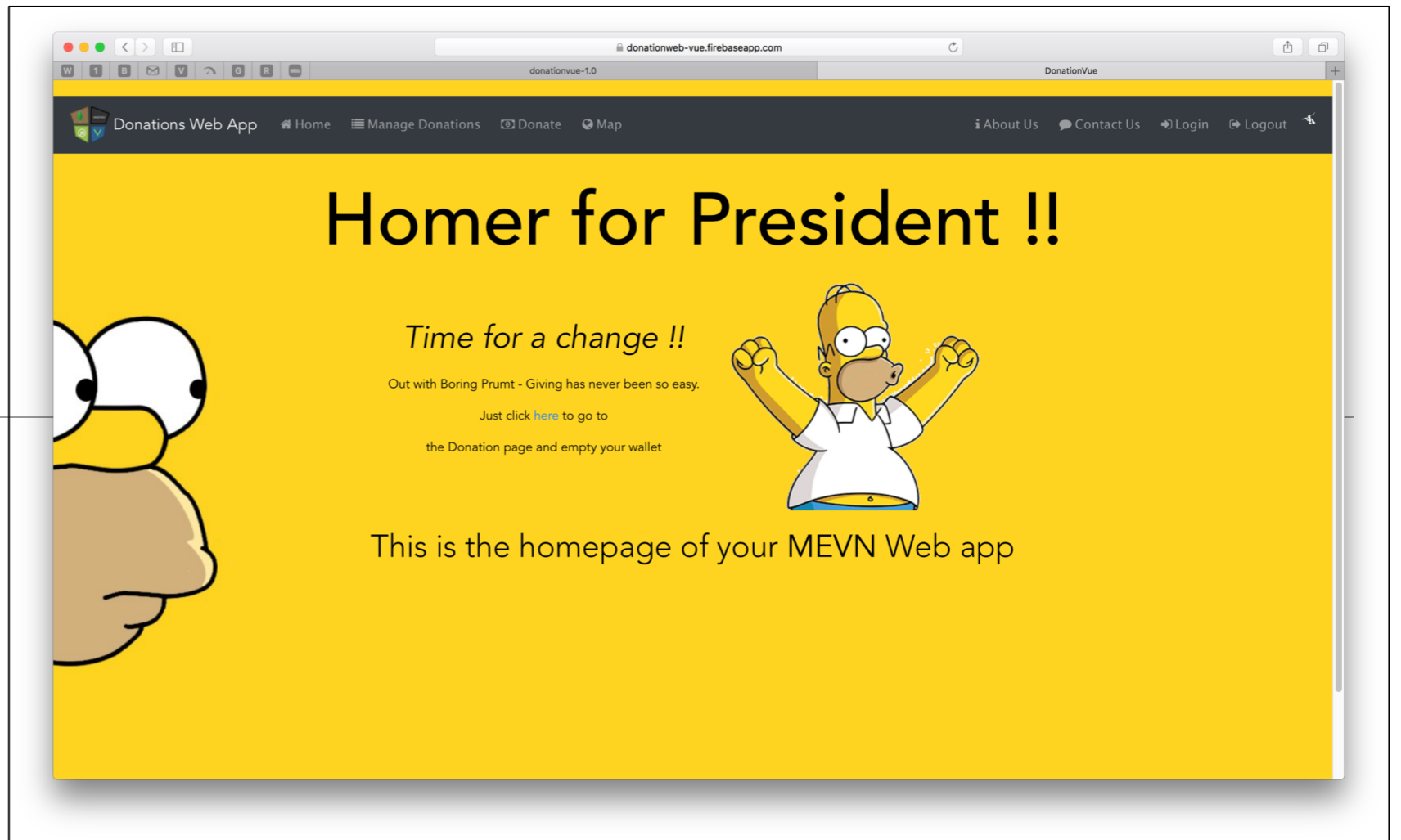


Assignment 2

70% of Overall Grade



Agenda

- Specification
- Grading Rubric
- Submission Guidelines
- Presentation / Video



Agenda

- Specification
 - Grading Rubric
 - Submission Guidelines
 - Presentation



Assignment 2 – Options

Continue working on your own app, exhibiting similar level of complexity/feature density as covered in the 2nd part of the Semester Case Study.



Case Study RECAP – Donation (Assignment 1)

- A Node Web Server to manage donations made to ‘Homers Presidential Campaign ’.
- App Features (all via RESTful API)
 - POST a payment type and donation amount in JSON format
 - GET a list of donation amounts and types
 - GET an individual donation using an ID
 - DELETE an individual donation using and ID
 - Upvote a donation via PUT request
- Persistence via MongoDB deployed to Heroku

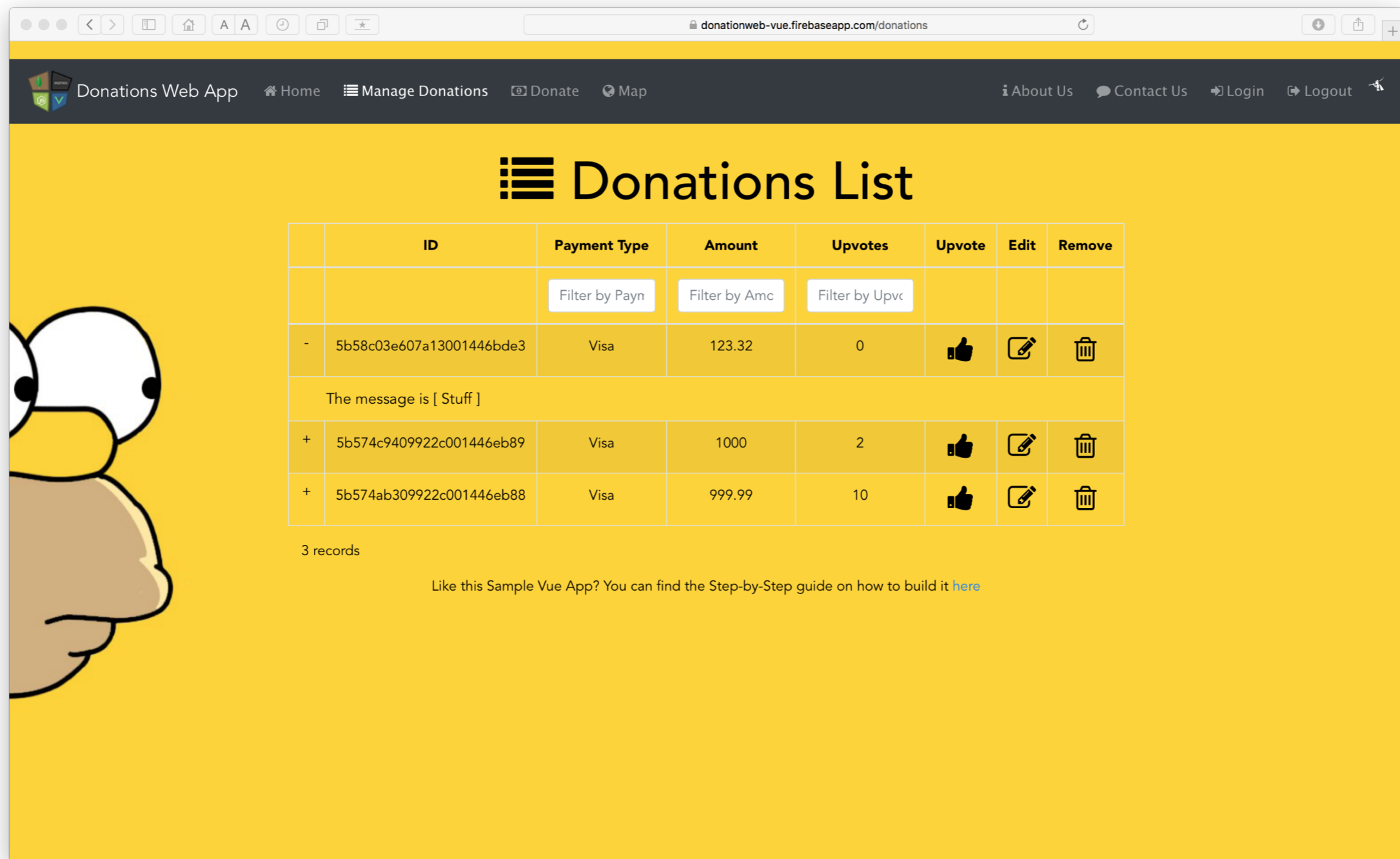
Case Study – Donation (Assignment 2)

- A FULL JS Web App with a Node Back-end and Vue front-end to manage donations made to ‘Homers Presidential Campaign’.
- App Features
 - Make a Donation
 - List / Filter / Sort All Donations (and show the most ‘upvoted’)
 - Upvote an individual donation using an ID
 - EDIT / DELETE an individual donation using and ID
- Persistence via MongoDB

Sample Extra Features

1. Enable User Signup / Registration / Login.
2. The donations are persisted (in a MongoDB database), and will be reloaded when a user logs in.
3. Support viewing/updating individual donations.
4. Allow a user to delete their own donations from the database.
5. Store a location with the donation and display on a Map, with donation info attached to marker.

Web App Features – List / Filter / Sort / Edit / Delete



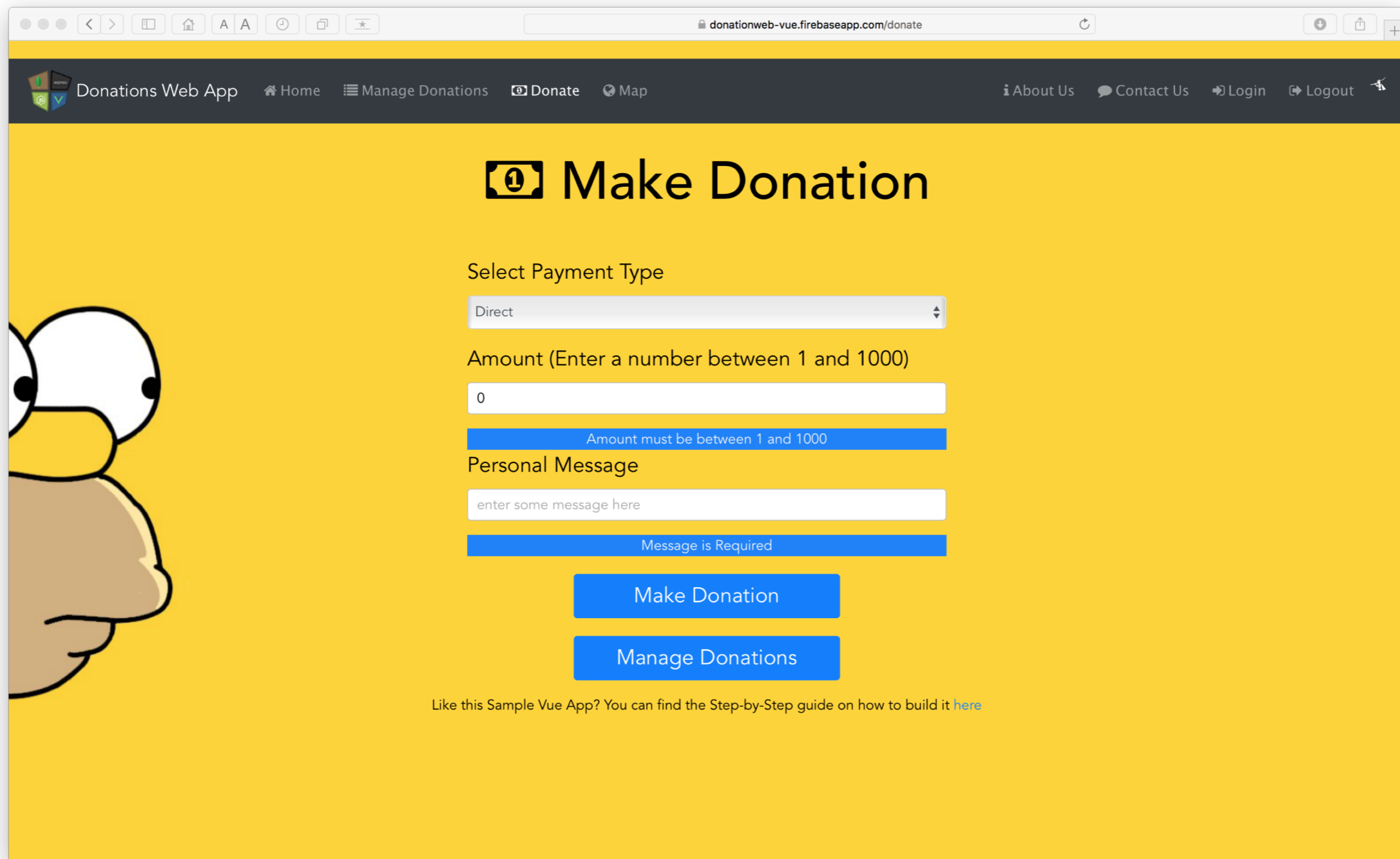
The screenshot displays a web application interface for managing donations. The page title is "Donations List". The navigation bar includes "Home", "Manage Donations", "Donate", "Map", "About Us", "Contact Us", "Login", and "Logout". The main content area features a table with columns for ID, Payment Type, Amount, Upvotes, Upvote, Edit, and Remove. There are three filter buttons: "Filter by Payn", "Filter by Amc", and "Filter by Upvc". The table contains three records. A message "The message is [Stuff]" is displayed below the table. A cartoon character is visible on the left side of the page.

	ID	Payment Type	Amount	Upvotes	Upvote	Edit	Remove
		Filter by Payn	Filter by Amc	Filter by Upvc			
-	5b58c03e607a13001446bde3	Visa	123.32	0	👍	✎	🗑️
The message is [Stuff]							
+	5b574c9409922c001446eb89	Visa	1000	2	👍	✎	🗑️
+	5b574ab309922c001446eb88	Visa	999.99	10	👍	✎	🗑️

3 records

Like this Sample Vue App? You can find the Step-by-Step guide on how to build it [here](#)

Web App Features – Make a Donation



donationweb-vue.firebaseio.com/donate

Donations Web App Home Manage Donations Donate Map About Us Contact Us Login Logout

Make Donation

Select Payment Type

Direct

Amount (Enter a number between 1 and 1000)

0

Amount must be between 1 and 1000

Personal Message

enter some message here

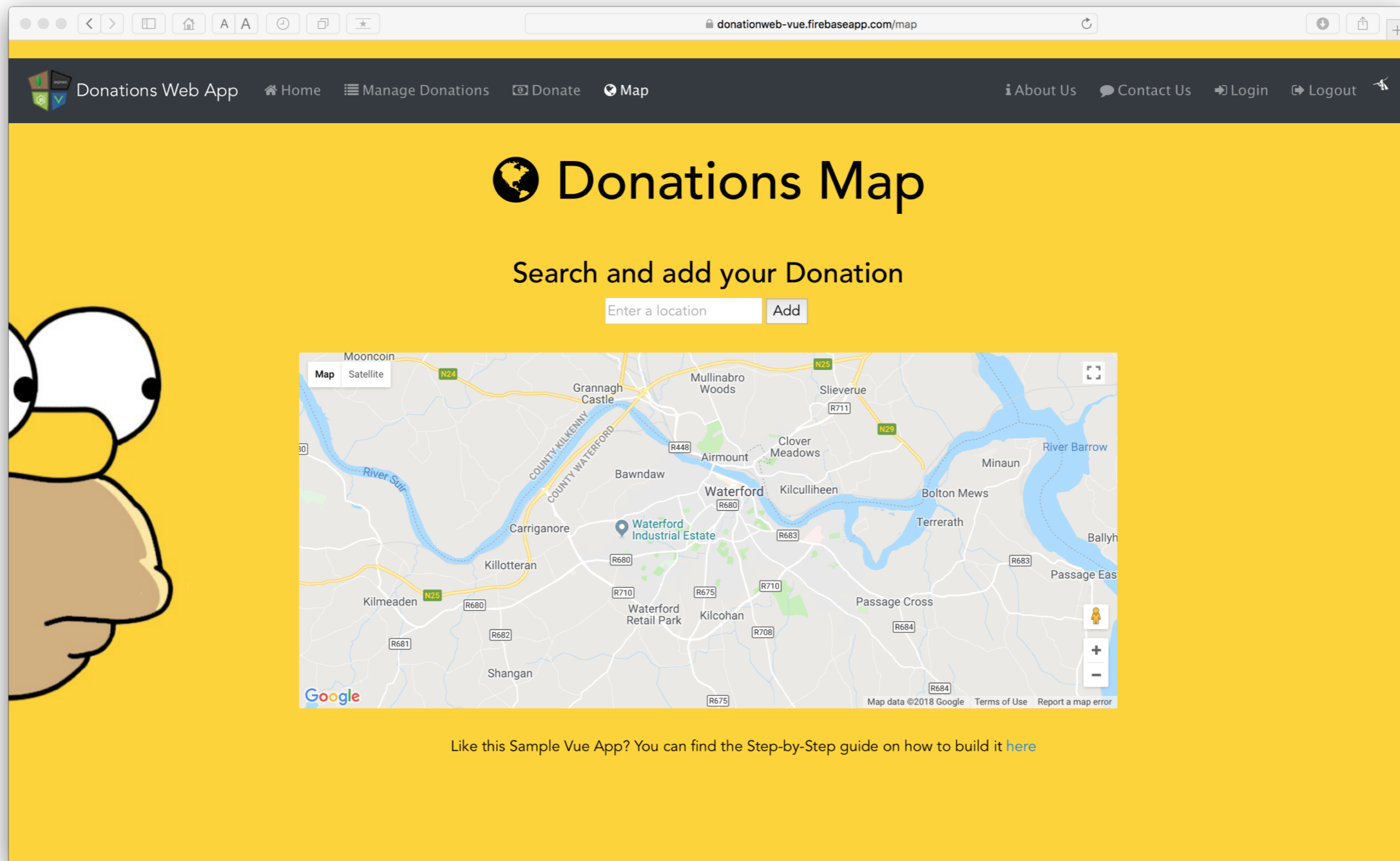
Message is Required

Make Donation

Manage Donations

Like this Sample Vue App? You can find the Step-by-Step guide on how to build it [here](#)

Web App Features – Map



The screenshot shows a web browser window displaying a web application. The browser's address bar shows the URL `donationweb-vue.firebaseio.com/map`. The application's navigation bar includes links for Home, Manage Donations, Donate, Map, About Us, Contact Us, Login, and Logout. The main content area has a yellow background with the title "Donations Map" and the instruction "Search and add your Donation". Below this is a search input field with the placeholder text "Enter a location" and an "Add" button. A map of the Waterford region in Ireland is displayed, showing various locations and roads. A cartoon duck character is partially visible on the left side of the screen. At the bottom, there is a link to a "Step-by-Step guide on how to build it here".

Donations Web App Home Manage Donations Donate Map About Us Contact Us Login Logout

Donations Map

Search and add your Donation

Enter a location Add

Map Satellite

Waterford Industrial Estate

Like this Sample Vue App? You can find the Step-by-Step guide on how to build it [here](#)

Agenda

- ~~Specification~~
- Grading Rubric
- Submission Guidelines
- Presentation



Assignment Rubric for Assignment 2

Standard	Client Functionality [60%]	Server Functionality [15%]	UX [15%]	DX [10%]
Baseline	Assignment 1 Functionality with Basic CRUD	MongoDB + Schema	App Navigation (via Menus)	Data Validation
Good Pass line	Additional Functionality as part of CRUD eg searching/filtering	> 2 Additional routes	Use of UI elements to complement UX eg DatePicker	Adherence to JS Best Practices eg SoC, Design
Very Good	Use of > 1 3 rd Party API	> 3 Additional routes + Additional Models	UI Guidelines adhered to	Automated Testing
Excellent/ Outstanding (70%+)	Use of > 3 3 rd Party APIs/ Google APIs	Cloud Support/acts as BaaS	Material Design Guidelines adhered to	Repo Usage, git etc.

Agenda

- ~~Specification~~
- ~~Grading Rubric~~
- Submission Guidelines
- Presentation



README / Design Document file

Include a DESIGN Document file (max 20 pages):

- Name and Student ID.
- Full description of Web App functionality, including, Server & Client, specific Frameworks used and if any, 3rd party and/or Google APIs used.
- Appropriate UML Diagrams & Use Cases
- Database Schemas
- Git approach adopted and link to git project / access.
- UX/DX approach adopted.
- References

Submitting Project Deliverables

Submit zip of project via Moodle dropbox. This zip should also include:

- The Design Document file,
- full source of your web project and
- Youtube link to video (5 – 10 mins MAX) of Web App Testing

Give read access to your lecturer to your GitHub / BitBucket repos. GitHub and BitBucket ids are:

- **ddrohan.**

Agenda

- ~~Specification~~
- ~~Grading Rubric~~
- ~~Submission Guidelines~~
- **Presentation**



Presentation / Video

You will be allocated a 15 minute slot in the week 12 practical labs to present your project.

- Attended by Tuition team only.
- 15 Minute to include demo + Q&A.

Note: I will be strict on the 15 minute allocation, so please arrive into the room with your Laptop ready to go with your app / code walkthrough.

Questions?

The logo for MEMN features the letters M, E, M, and N in a bold, white, sans-serif font. The second 'M' is stylized with a dark blue center and green outer sections. The background is a horizontal gradient from purple on the left to green on the right.

MEMN

MONGODB - EXPRESS - VUEJS - NODEJS