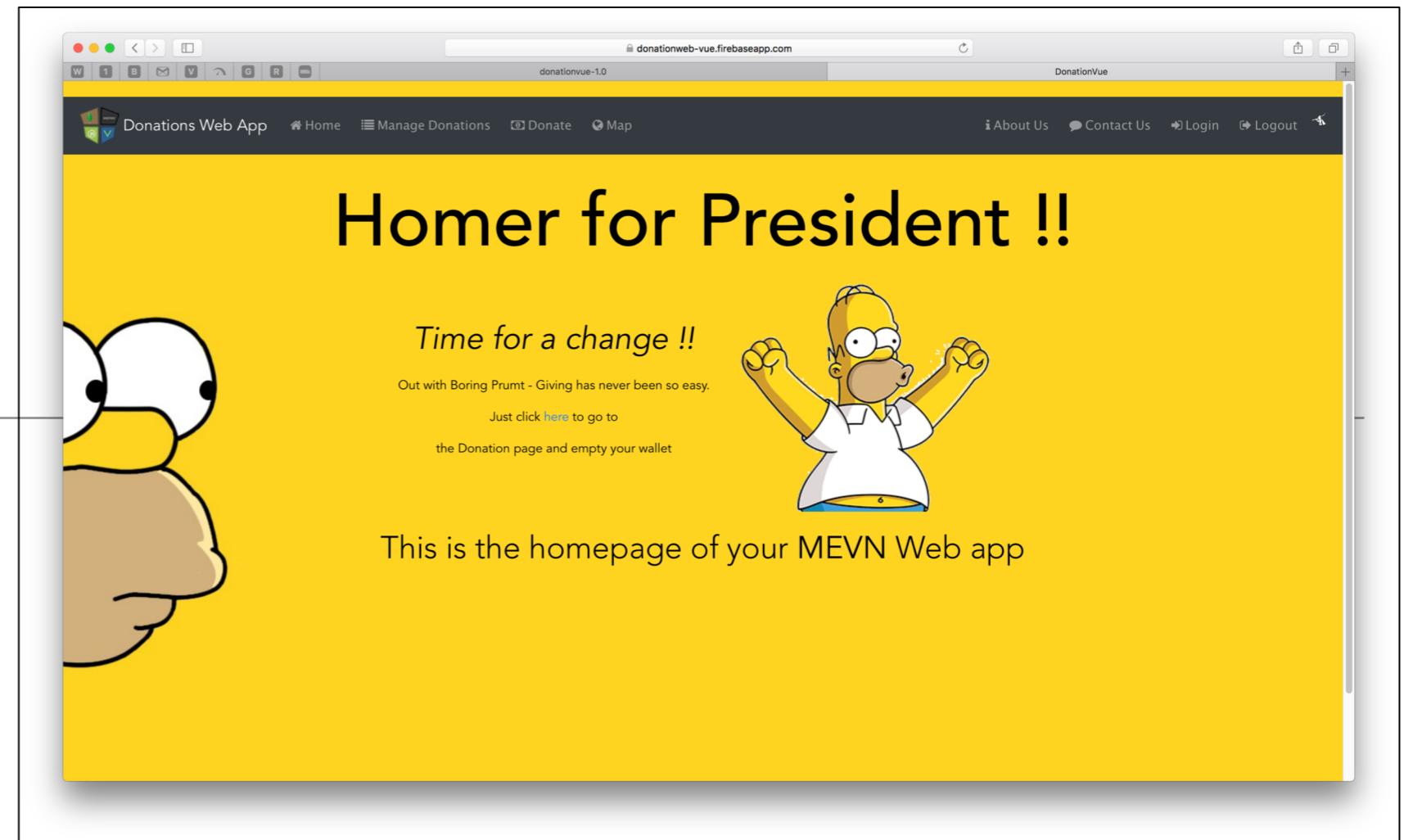


Assignment 2

70% of Overall Grade



Agenda

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



Agenda

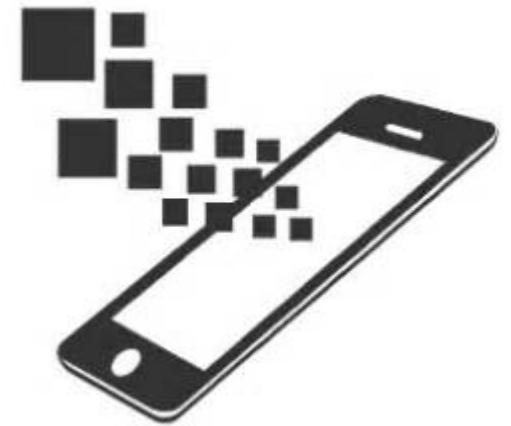
- Specification

- 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

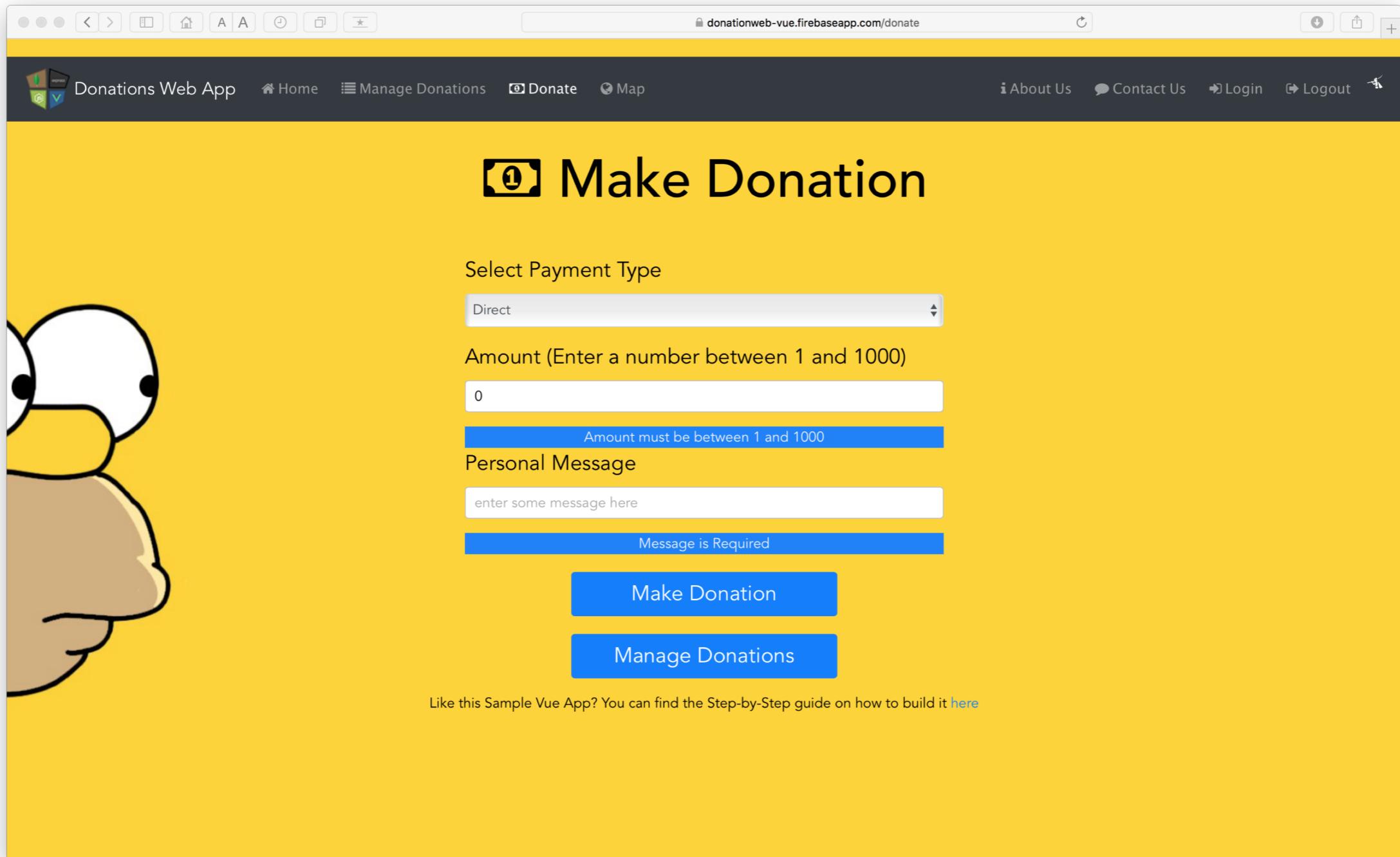
The screenshot shows a web application for managing donations. The URL in the browser is `donationweb-vue.firebaseioapp.com/donations`. The interface includes a top navigation bar with links for Home, Manage Donations, Donate, Map, About Us, Contact Us, Login, and Logout. A large yellow sidebar on the left features a cartoon character's head. The main content area has a yellow background and displays a table titled "Donations List". The table has columns for ID, Payment Type, Amount, Upvotes, Upvote, Edit, and Remove. It includes filter buttons for Payment Type, Amount, and Upvotes. The data table shows three records:

	ID	Payment Type	Amount	Upvotes	Upvote	Edit	Remove
-	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



The screenshot shows a web application interface for making a donation. The top navigation bar includes links for Home, Manage Donations, Donate, Map, About Us, Contact Us, Login, Logout, and a search icon. The main title is "Make Donation" with a dollar sign icon. A large yellow background features a cartoon character's head on the left. The form fields include a dropdown for "Select Payment Type" (set to "Direct"), an input field for "Amount" (containing "0") with a validation message "Amount must be between 1 and 1000", and a text area for "Personal Message" with a validation message "Message is Required". Below the form are two blue buttons: "Make Donation" and "Manage Donations". At the bottom, there is a link: "Like this Sample Vue App? You can find the Step-by-Step guide on how to build it [here](#)".

Donations Web App

Home Manage Donations Donate Map

About Us Contact Us Login Logout

\$ 1 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 the 'Donations Web App' map feature. The URL in the address bar is donationweb-vue.firebaseioapp.com/map. The page has a yellow header with the app's logo and navigation links: Home, Manage Donations, Donate, Map, About Us, Contact Us, Login, Logout, and a search icon.

The main content area features a large title 'Donations Map' with a globe icon. Below it is a subtitle 'Search and add your Donation' with a search input field and an 'Add' button. A Google map of Waterford, Ireland, is centered on the city. The map includes labels for 'COUNTY KILKENNY' and 'COUNTY WATERFORD'. Key locations marked on the map include Mooncoin, Grannagh Castle, Mullinabro Woods, Slieverue, Clover Meadows, Minaun, River Barrow, Waterford Industrial Estate, Carriganore, Killotteran, Kilmeaden, Shangan, Bawndaw, Kilcohan, Waterford Retail Park, Passage Cross, and Bolton Mews. Roads labeled include N24, N25, R448, R680, R681, R682, R683, R684, R685, R686, R687, R688, R689, R690, R691, R692, R693, R694, R695, R696, R697, R698, R699, R700, R701, R702, R703, R704, R705, R706, R707, R708, R709, R710, R711, and R712. A sidebar on the left contains a cartoon illustration of a person's head and shoulders.

At the bottom of the page, there is a link: '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 3rd 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 consists of the letters M, E, V, N stacked vertically. The letter M is white on a purple background. The letter E is white on a blue background. The letter V is composed of two overlapping triangles: a green one pointing down and a dark blue one pointing up, both set against a teal background. The letter N is white on a green background.

MEVN

MONGODB - EXPRESS - VUEJS - NODEJS