
Final Year Project Title

Aaron Moran

Conor Shortt

Thomas Kenny

B.Sc.(Hons) in Software Development

APRIL 20, 2021

Final Year Project

Advised by: Kevin O'Brien

Department of Computer Science and Applied Physics

Galway-Mayo Institute of Technology (GMIT)



Contents

1	Introduction	4
2	Context	5
2.1	Filler	5
3	Methodology	6
3.0.1	Testing	7
4	Experiments	8
5	Results	10
6	Conclusion	11
7	Technology Review	13
7.1	XML	13
8	System Design	14
9	System Evaluation	15
10	Conclusion	16

About this project

Abstract This project is an Social Media platform where you can create communities, post up discussions and follow your friends. . . . etc etc

Authors Aaron Moran, Conor Shortt, Thomas Kenny.

Chapter 1

Introduction

The final year project requires you to show initiative, consistency and the ability to showcase what you have learned over the course of four years. Three members consist of Aaron Moran, Conor Shortt and Thomas Kenny. We are like minded students that share an interest for learning new skills and developing our ability as computer programmers. We have different skills that when combined complement each other from creativity to problem solving. With our interest in social media we decided early on that a social media website was a project that we felt we all could contribute too. A social media website requires multiple features, the in depth use of a database and an engaging UI. We were confident from the start with the team we have we could develop a website that could meet the standards of the websites out in the world today. Before any developing took place, it was crucial we carried out in depth research into the top social media websites , how we were going to develop it , type of database to use and too draw up a timeline of goals/targets.

Chapter 2

Context

- Provide a context for your project.
- Set out the objectives of the project
- Briefly list each chapter / section and provide a 1-2 line description of what each section contains.
- List the resource URL (GitHub address) for the project and provide a brief list of the main elements at the URL.

2.1 Filler

Chapter 3

Methodology

About one to two pages. Describe the way you went about your project:

- Agile / incremental and iterative approach to development. Planning, meetings.
- What about validation and testing? Junit or some other framework.

For the development of this application, we decided the best approach to take was to use Agile. Agile is a methodology focused around iterative development. Using this method requires a team to work collaboratively and consistently. Before we started any development it was important to have a clear plan of what the features we wanted to incorporate and develop. Living in the same house proved useful in these times as we were able to have in-person meetings. Our first meeting we drew up a plan of how we were going to develop this application. Our first aim of was to agree on a development stack. After research into numerous development stacks such as the MEAN stack and MERN stack we decided on the MERN stack. Our research included ease of use , package management and plentiful of resources at our disposal. With our choice of stack agreed upon, the next step was dividing the roles in the group. Conor and Thomas have an interest in back-end development and Aaron was keen on developing the Front-end. The first meeting was beneficial as we had a concrete foundation to build upon.

We decided the best approach to development was to work on two - three features a week. We set up a whiteboard with a list of features to be developed, every week we would divide the features between the three of us. An example of our work process is shown below. ===IMAGE OF WHITEBOARD=== Each week we would have a meeting

detailing the work done and evaluate the features developed. This was an important process as we got to discover what we could improve and what works well. If something needed improving we would put it back on the list for future meetings. If a feature was functional and proved to be efficient we would cross it off the board.

These meetings were also valuable for tracking bugs throughout the development process. If a team member encountered a bug or an issue we would write the it on the board and discuss it at the next meeting. Bug tracking was a process that requires good communication and teamwork as for an issue to be overcome we had to share solutions and potential bug fixes.

Throughout the development of this project, we have used many collaboration tools including GitHub, Slack and Discord. While using these tools we were able to communicate new ideas and solutions to problems we may have been facing. GitHub helped us to track our progress and any mistakes that were made we could easily rollback commits. As we are all working on this project remotely, VOIP was incredibly important as we could communicate instantly, raising any issues we may be having and coming to brainstorming a solution.

3.0.1 Testing

Testing was a vital part of the development process. Using Agile required us to test after each feature was incorporated. We used many testing methods during the process of this application the two main testing methods used are White box testing and Black box testing.

- Selection criteria for algorithms, languages, platforms and technologies.

Check out the nice graphs in Figure 6.2, and the nice diagram in Figure ??.

Chapter 4

Experiments

We tried out many of different experiments throughout the development of our project to improve its user experience and run-times. Between Front End and Back End we tried out many different design patterns and user interfaces that would suit our project and through trial and error found the best suitable solutions to suit our needs. We were able to utilize Googles lighthouse performance testing tool to examine our load times.

We faced issues while trying to load a large amount of images on a users feed and decided to experiment with ways around this bug. We tried reducing the amount of imported packages and keep the project package small so that it would be less pressure on the website when running. This clean up didn't seem to make any improvements on our website speeds and load times, but was still a successful experiment as it tidied up our code and removed any unnecessary and unwanted external packages.

We then attempted reducing the amount of shared information between collections within our MongoDB database, instead of passing all of the posting users information we only shared the ID keys which would give us full access to the details of the posting user and the posts information. Changing this structure within the communication between our collections saved us many resources and storage space within our database. This only marginally improved load times but was a success as it made our website perform at a higher speed than previously, and we learned a new solution to sharing information within our application which could be applied to many other features and functionalities.

AMAZON SOLUTION HERE

- Website Runtimes
- Responsiveness
- Speeds and Performance
- Loading in images via MongoDB/Amazon
- Front End
- Colorways
- UI
- FIGMA, WIREFRAME
- Design Decisions
- Icon imports, Bootstrapping

Throughout the development of this project we tried out many different User Interfaces and Front-End designs to try and find which was the most suitable layout for our project. We used software such as Figma and Wireframing to design and test our designs which we could implement, once we had three official designs we came to a conclusion on which one would be most suitable for our project to match its features.

Chapter 5

Results

Chapter 6

Conclusion

Ultimately, working as a team to develop this project etc etc ...

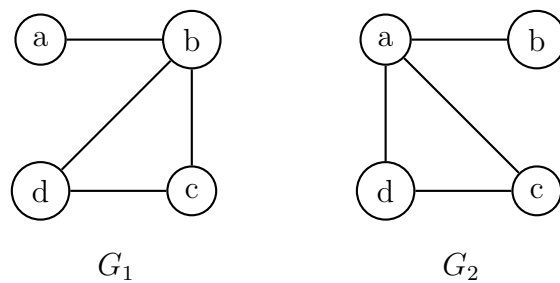


Figure 6.1: Nice pictures

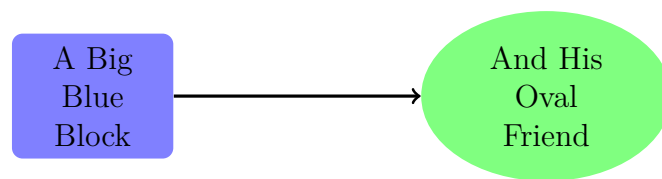


Figure 6.2: Nice pictures

Chapter 7

Technology Review

About seven to ten pages.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XML, WSDL, JSON, JAXP. The development stack we have chosen is the MERN stack, this consists of MongoDB, ExpressJS, ReactJS and NodeJS. The reason as to why we chose this development stack is because it can be easily divided up between Front-End and Back-End. Although we cross collaborated on both sections and helped each other whenever we were facing any issues, it was easiest to give each team member a main focus. Aaron worked mostly on the Front-End and Thomas and Conor worked together on the Back-End.
- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

7.1 XML

Here's some nicely formatted XML:

```
<this>
  <looks lookswat="good">
    Good
  </looks>
</this>
```

Chapter 8

System Design

As many pages as needed.

- Architecture, UML etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

Column 1	Column 2
Rows 2.1	Row 2.2

Table 8.1: A table.

Chapter 9

System Evaluation

As many pages as needed.

- Prove that your software is robust. How? Testing etc.
- Use performance benchmarks (space and time) if algorithmic.
- Measure the outcomes / outputs of your system / software against the objectives from the Introduction.
- Highlight any limitations or opportunities in your approach or technologies used.

Chapter 10

Conclusion

About three pages.

- Briefly summarise your context and ob-jectives (a few lines).
- Highlight your findings from the evalua-tion section / chapter and any opportuni-ties identified.

Bibliography