Pitch –

Hello, we are team 29, we are pitching for the Smart Receipts project, and I am--

[Introduce Lead 1, whoever you are, and your role, and then the main roles (admin, team lead, git lead)]

When approaching this project, we first needed to understand the underlying issue that this project was trying to solve. It's easy saying that people have impulsive habits when spending on their grocery shopping, but the extent was unknown to us.

To a great degree, many can generalise spending too much on things you don’t need as a simple fix and novel. But, when we found that households (on average) spend about three times their disposable income on non-essentials, it fell that the issue was of a much wider prevalence and scale than one may assume.

Intuitively it can be hard to grasp the growth of small costs, and groceries are the epitome of small costs piling up. The idea of easily inputting a grocery shopping list into a system that can visualise and categorise the areas of unneeded spending, and culminate them over periods, grew in importance.

Fundamentally, there are two solutions to tackle someone’s detrimental spending. One would be to show what they are spending on things they don’t need (and can cut back on). This can be akin to showing a smoker a picture of their lungs. For some, this may be enough to kick them into improvement.

However, realisation does not conjure action. Action comes from an active ideal of improvement, and for some, it may be difficult to come up with a strategy to solve their overspending. So, there comes the idea of having an action plan thrown at them. A plan that can ease them into better choices, that can stop those who may think doing extreme changes in spending would not have a high probability of failure.

To have both the plan, and the issue; it would be difficult for the user to not at least attempt it, and if that fails, then the scope of the problem will have exceeded what an app can solve.

There are existing solutions that partially implement either solution. For example, SmartReceiptsLLC contains the ability to scan receipts, but it by default does not extract any information from the receipt. It allows OCR, but this is a paid OCR using the taggun API.

It can create reports but does not separate costs into essential and non-essential. This may be a useful tool to see how much you spend overall, but it is limiting in seeing what part of that can be improved.

So, a solution of our scale would actively improve the space of managing your spending on groceries. This improvement will coincide with your core values as a company, like improving credit scores. It is hard to improve your credit score from a low point, and for many the very task to start is daunting, so we believe in the importance of developing accessible software that requires as little user input as possible.

To make something as user-friendly and interactive as possible – that ideal and shared interest with you is our motivation.

On the technical side of things, there are two main methods of approaching the problem of an interactive mobile program. Either we make a native app or a portable web app. Native apps offer better functionality due to them being tailored to a system, either Android or IOS, but the downfall we found was that we would need two codebases built entirely in different languages to reach the widest possible audience.

With the time we have, a native app would have to be developed only on android. So, naturally, we looked at the space of websites. Websites have been a growing space for a long time, and the introduction of modern web frameworks, and faster browsers, allows them to creep into near-native performance.

Not just us have realised this. Many major companies have taken to transforming websites into web apps, treating them as native applications using various approaches, such as Electron for desktop, modifying Service Workers for offline functionality, and even using Native versions of popular frameworks like React Native.

A Progressive Web App is a form of moving to web technologies by improving offline functionality and utilising caching. Spotify and Uber use this as a few examples. It streamlines development into a well-documented space, and for us the advantage of portability is important – the main goal is to reach as many vulnerable customers as possible.

The other advantage is that we can allow desktop management of their goals. On any platform, we can procure some solution without redefining the entire codebase.

Now, non-native web apps will have their disadvantages on phones, but you just need a fast framework to remove extra battery utilisation and inefficiencies. Thus, after a bit of perusing, we found that Svelte, being one of the most popular upcoming web frameworks, is a good pick.

It works on a system of pre-compiled frontends, removing any non-native jargon from the system. This greatly improves speed, which we see as a requirement to allow a wide breadth of phones to run it.

The backend will be a combination of node.js, specifically express.js as a wrapper, and some implementation of a neural model, typically in python.

The greatest unknown in the project is how to properly implement a model that can first do reliable text detection on the receipt, and then an OCR model that can translate it to usable text to manipulate. We aim to use frameworks like PyTorch and cross-reference research papers on the subject.

MySQL will be used as the storing method for user data, which will be interfaced in the node.js backend.

Now, it's good we have an idea of what to do, but the skills of our team need to apply to the project.

We have all made a PHP website, and specific members have further experience working with front-end frameworks like Material UI to create modern styling. Things like making a countdown website to sort large amounts of data locally.

Our team also has experience in UI development. We have used Java to create basic game UI, and Alfie has experience in Lua creating complicated UI systems for RPG-style games and WordPress.

We have interfaced with MySQL with PHP and understand practices to efficiently and securely store data. Our team has experience running web servers and managing them via Google Cloud and Digital Ocean via setting up VPNs, WordPress websites, and open-source projects to mess around.

For AI, we have all used python with libraries such as pandas, and NumPy to manage and manipulate datasets. We then used scikit-learn to train predictive models, including the 3 main neural networks, decision trees and linear regression models. Alan in our team also has more experience outside academia, building an AI neural network in C++ from the ground up. We have also used python to interface with Raspberry Pi.

Most of us are taking the 2nd year module on AI, so we are prepared to learn on this project.

We know we will have to learn a lot during this project, but we have a stable foundation to propel that learning efficiently. We are motivated to learn modern web frameworks and attempt a useful but difficult AI solution – this project is an exciting mix of problems to solve.

Our team will approach the project using Scrumban.

To summarise:

We understand the reasoning behind this project and can empathise with the growing struggles over the current years for many people. There exist solutions in the app space, but they are lacking either in data visualisation capability, or streamlined receipt input.

Our experience concerns PHP, Javascript, and of course HTML and CSS through either in-course website projects or extracurricular/hobby undertakings. This includes a countdown solver website using Material UI, and PHP database management projects using modern CSS styling.

Our MySQL experience concerns varying levels of manipulating and ensuring it's secure and efficient. We have experience in server management via Google cloud and Digital Ocean and have python usage outside academia in AI.

We hope to approach the problem with a website framework, through the fundamental of easy portability.

We hope you enjoyed our pitch.