PORYGON GROUP PRESENTATION

(we can each talk for a part of the presentation I think)

[Shani speaks]

[Slide 1] Hello everyone, and welcome to our sprint 2 demo! [everyone cheers 😍🔥❗💯]

Let's begin with a short presentation on our **project progress** and what we've been up to.

[Slide 2] Our objective for this project is to provide an Interactive Web Interface for developing, back testing, and analysing FOREX algorithmic trading strategies. In addition to this, our product also integrates data visualization, anomaly detection, and version control.

[Slide 3] Now that you know our goal, let's introduce **our team members**, and a bit of what they did!

(Everyone says their name and very short what they did)

"I'm Shani, and I worked on displaying the graph, creating a CI/CD pipeline along with some tests, and the presentation."

"Hi I'm Jamie and I did the anomaly detection, as well as working on the presentation" [Change this as you want I just made it up]

"Hi I'm David and I did the code-server and run functionality"

"Wazzup I'm Shalomé …"

"I'm Amir and I have done some of the frontend for the editors page as well as input validation."

[Jamie speaks]

Before we get into our actual progress, let's just show you a bit on how our **branching** worked for this sprint.

[Slide 4] This is our **git branch setup**!

Let me go through what's going on here step by step for you, so that it's less confusing. Let's start from master.

From the master branch, 3 feature branches were created.

These branches were:

- Shani-graph, where the graph feature was added.
- Jamie-smith, where the anomaly detection feature was added, and
- David-code-server-new, where the vscode server feature was added.

Ignoring the names of these branches, we move on to the branches that were branched off the previous three.

- Shalome-metrics was branched off Shani-graph. This branch implemented the feature to choose which output data to display.
- Amir-website was branched off David-code-server-new. This branch added the run functionality for the backtesting.

Now you understand the layout, let me lastly just say which branches were merged back into our main branch. The branches merged were Shalome-metrics, Jamie-smith and David-code-server-new(since amir-website was merged back into this branch)

[David speaks]

[Slide 5] Moving on, let's talk a little about our **key accomplishments and deliverables for sprint 2**!

On the backend side, we had David, Amir, Rohan, and Jamie working together to get this project working!

This included git integration for version control and anomaly detection to make it possible to spot anomalies! You can now also run backtests and display the output, which is up next as…

Frontend! On the frontend side, Shani, Jamie and Shalomé worked together to get the visuals you'll see in a bit. There, an archive page was added, though its name is just Metrics, for data organization, as well as a graph to visualize the output data received from running a backtest! We also added local version control for improved user experience and workflow management.

Last but not least, we used Shortcut to plan and assign each task so that the work is spread out evenly.

[Shalomé speaks]

[Slide 6] Now it's time to describe our **agile process** for this sprint.

Firstly, *Sprint Planning*!

Planning this sprint included defining the sprint scope and prioritizing tasks and user stories. We also discussed the design of the user interface and broke each tasks into clear, manageable units.

Secondly, *Incremental Development and Iterations*!

Using incremental development allowed us to gradually enhance features and refine this sprint. Iterations provided feedback and change, improving the functionality and user experience of our product.

[Amir speaks]

I'll talk a little about our *Backlog Refinement* and *Continuous Integration and Testing*!

Our product's backlog was regularly updated with new features and improvements, to ensure our product includes everything it needs to.

In addition to this, bug reports and technical debt were tracked in the backlog for resolution in the upcoming sprints.

[Slide 7] In terms of continuous integration and testing, our code was frequently integrated into the main branch, ensuring stability with each update. We used tools like GitLab CI/CD pipelines, which facilitated rapid and secure software releases.

[Shani speaks]

Lastly, I will talk about our *Minimum Viable Product,* or MVP, and the delivery of it.

Our MVP features essential functionalities, such as:

- Git integration, which was used for version control,
- a Code editor, so the user can specify their own strategies,
- a Metrics page for the visual representation of trading data and performance, and
- a sample of what's to come for the next sprint: Anomaly detection! I will let Jamie explain what that is. [Jamie explains]

[Slide 8] That's all for our presentation, thank you for listening. Now let's move on to the fun part: the live demonstration!