

# MordOS

## Quick tech brief

You'll be building a single-page application, so no need to bother with databases, BE environments, dockers and similar. The application should be easily buildable and runnable from the source files which should be available from a Github repository (private via access) with complete history.

## Intro: The one OS to rule them all

You have just received a Slack message from Pete, our salesperson.

Borgoth, a serial entrepreneur with a plethora of mostly failed ventures, has contacted us to work on his new idea. You can find a short brief below:

Mord OS, the operating system to rule them all. Borgoth believes that most entrepreneurs and managers in his line of work require a more tailored operating system. It is possible to work with all the different tooling that is already available but the constant context switch is not productive and staggers the flow of work. He firmly believes that it should be web based so it is virtually available everywhere. At this stage it would serve as an internal tool for his company but he is willing to offer it later, if it receives traction.

## Product development: Mord OS

Your assignment is to create a graphical user interface for a web based operating system. That couldn't sound more wild than it actually is. **This is an MVP** so functionally it'll be quite far from an OS but it does need to **look and feel like an OS, be appealing and fun to explore**.

Borgoth, being the flaming venture demon, completely understands a lean approach and gives us our full support. However, he believes some **functionalities bring a lot of value**:

MordOS **should support a directory mechanic for managing folders, files and applications**. Because Morgoth is also an avid writer he needs a **plain text editor** to

bang out his novels.

There should be a simple authentication feature so not everyone can access the OS. A simple email and password flow will do for now. (*email: [borgoth@mordos.com](mailto:borgoth@mordos.com) / pass: 12bindthem*)

Because Borgoth considers himself a visual person, the OS should support **some form of customisation and configurability**. He also works late into the night a lot so an ability to easily switch between day and night modes would be very helpful to him.

Borgoth hasn't decided on the look and feel of it yet and he doesn't want us to reinvent the wheel. The OS should **look familiar to other systems** so it doesn't create confusion.

**Other functionalities that are not as crucial** but could really help enrich the OS:

To keep up with all the news and banter going on, the OS could include a dedicated RSS reader app. (*data source: <https://jsonplaceholder.typicode.com/comments>*)

Having a camera app seems like a must for any OS today and Borgoth is a huge fan of 80s filter on Instagram.

It would be good if the OS had a Gallery app to view local images and images from different sources without having to switch. (*data source: <https://jsonplaceholder.typicode.com/photos> , but can include others*)

The OS could also have a dedicated web browser. Can be simple at first.

With Borgoth often working late into the night he sometimes deletes files by mistake, so he believes that the **OS should have a fault-tolerant and Borgoth proof file removal system** that would help him mitigate these errors and recover when they happened.

Borgoth isn't limiting us on anything here: **You can enrich the experience as much as you want!**

Borgoth also mentioned that the project should be **well structured** and be **easily extended and scaled** once everyone sees that using Mord OS is the one true path, and the dev team will need to grow quickly.

# Divisions

## Basic

Rules:

- React, Angular or Vue framework
- no component UI libraries (Bootstrap, Material, etc.), no Tailwind
- the only UI framework allowed is ReachUI (or vue, angular alternative)
  - for Vue or Angular alternative confirm first with cadre
- styled components, sass, css modules are allowed
- predefined icon pack allowed
  - Feather
  - Iconic.app

These should cover your needs. If you want to use something else confirm with cadre.

- TS use is optional
- create-react-app, nx project bootstrap or other similar tools are forbidden
  - simple webpack, vite or similar setup is expected
- deployed to Netlify or a similar service

## “Hard” mode

Rules:

- no JS framework allowed
- no component libraries allowed
- only JS, CSS, HTML are allowed, optionally TS is allowed
- simple webpack, vite or similar setup is expected
- deployed to Netlify or a similar service

Hard mode assignments will get additional points for the extra imposed limitations.

The choice of the division is totally up to you. When registering please let us know your choice and again when you submit your task. The initial choice is not an obligation.

---

## Criteria

We will be grading the assignments in the following areas:

- UX
  - look and feel of the application
  - user flows
  - responsive design
  - etc.
- a11y
  - semantic HTML
  - taking colour and animations into account
  - etc.
- technical solution
  - developer experience
  - use of native browser APIs
  - code and file structure
  - use of proper patterns (that make sense for the use case), separation of concerns, proper use of JS or TS (if you decide to use TS)
  - consistency and reliability of code

- fun and innovative features

---

## Delivery

Submissions will be open **until 21.12.2022 23:59**. Submission **should include git repository** with a complete git history, **live environment** hosted on Netlify or similar and a **bullet list of things you'd like to high light or what you focused on**. If you produce any supporting material like estimations, diagrams etc. feel free to attach them to the submission.