FE Code Challenge (lite version)

The project focuses on creating a web application (inspired by the real Aspire application) divided in 2 sections:

1. CSS Challenge:

The candidate should convert the designs in an application with focus on design fidelity and usage of libraries and tools aligned with the Code Challenge.

All images assets are exportable directly from the links.

- Mobile View: https://xd.adobe.com/view/80c753f2-db2f-4dfc-b6c2-ce39a4c787f0-d594
- Desktop View: https://xd.adobe.com/view/80c753f2-db2f-4dfc-b6c2-ce39a4c787f0-d594/screen/7d8639be-16a2-4dc6-9ddc-c3bcd8d8f1ee/ (optional)

```
https://xd.adobe.com/view/80c753f2-db2f-4dfc-b6c2-ce39a4c787f0-d594
```

2. Code Challenge:

The task is to make the UI built with the CSS challenge dynamic and interactive.

General notes:

- application can be built with any tools / libraries (if needed)
- application should use an API architecture, where dummy data can be returned from the APIs directly within the JS code, no real server side code is required
- the data can be stored in localStorage or stage management system
- on the start up of the application, some cards should be already added by the system without need to prepare a layout without cards

Interactions needed for the challenge:

- Add new card:
  - open a modal/page where the user can add the card name and submit the new card
  - the expiration date and card number will be randomly generated by the system.
  - the card will be appended in the carousel together with the other cards
- Freeze/Unfreeze card
  - on click of the freeze card the card will get the status of frozen
  - the frozen card will look different from the others (UI not defined by the challenge and up to the candidate, but it can be a simple opacity applied)
  - the freeze button will change label in unfreeze for the frozen card
  - allow the possibility to rollback the card state clicking again the same button and unfreezing the card
- Cancel card
  - on click open a modal where a confirmation action is asked
  - remove card from the list

Must have

- **Mobile version** of the CSS challenge
- Form validations
- A link to a public repository hosting the code (ex. GitHub/Bitbucket/GitLab)
- A README.md containing all the informations that the reviewers need to run and use the app

Recommended

- Desktop version of the CSS challenge with responsive UI
- High design fidelity
- SCSS
- BEM or similar patterns
- Aspire is using **VueJs 3** with **Quasar** framework. The same technology is much appreciated but not mandatory.
- A link to a hosted version of the app (e.g. using Netlify)

Nice to have

- Pixel perfect fidelity
- TypeScript
- Interactions with elements (e.g. mobile scroll behavior as design, collapsible cards, debit card carousel etc.)
- Using CSS framework with relative helpers
- Unit tests