# Frontend Task

**1. Implement an Ionic 6 (based on Angular 13) application with the following features →**

- The freshly started application should contain 5 initial transfers.

- The homepage should be a list view of transfers as a angular grid table with the columns:
    1. Account holder
    2. IBAN: use the npm package *angular-iban* to format it
    3. Date: use date format (DD.MM.JJJJ)
    4. Amount: use German currency format e.g. 1.111,11 €
    5. Note: Long text

- Make the table sortable by date and amount.

- Provide a filter facility that filters the list by account holder and note.

- Provide operations to add, edit or delete list items.

- If the user clicks on the button “create”, show a modal with a form, that lets the user provide all properties except for ID. On saving create a new ID.

- If the user clicks on the button “edit”, show a modal with a form, that lets the user edit all properties except for ID.

- If the user clicks on delete, prompt for confirmation

The frontend shall validate the entries →

-- Date: not empty, German date format, instant in the future
-- IBAN: not empty, valid IBAN according to angular-iban
-- Amount: not empty, max 8 characters, min 2 characters, max 2 decimal places, only numbers, "." and "," allowed, min 50 €, max 20.000.000 €
-- In case of validation errors, show a proper error message.

**2. Create endpoints using NodeJS to handle transfer objects (POST, GET, PUT, DELETE).**

The transfer objects should have the following properties →

- ID: UUID
- Account holder: string
- IBAN: string
- Amount: Floating Number e.g. 10.99
- Date: JJJJ-MM-DD
- Note: Text

**3. Requirements**
- Try to use Ionic components
- Use NgRx
- Write Unit tests with Jasmine (Code coverage should be over 80%)
- Write at least one e2e test with Cypress