

Question 1:

For my encoding and decoding of the code I decided to convert a numeric representation of the information into base32. The numeric representation is set up into 13 characters, the first 3 represent the store code, the next 6 store the current date of creation, and the final 4 show the transaction number of a given day. Example

Store: 45, Day 08.12.21, Transaction 9213

Numeric Code: 0450812219213

I then take this numeric code and turn every 3 characters it into base32 which will result in 9 characters remaining. An example of this is shown below again:

Numeric Code: 045 | 081 | 221 | 921 | 3

Base32 Code: 1D | 2H | 6T | SP | 3

Finally, to make it harder to cheat the systems I offset the numeric code before conversion by adding 24 to the numeric chunks. This means when converting 045 we first add 24 and then convert to base32. This number was chosen as it will take the largest possible chunk (999) to the maximum 2 character limit of the base32 representation(1023). This helps to prevent people from trying to find valid codes by making it significantly harder.

For decoding, I reverse all the steps outlined above to get the original numeric representation of which the information can be extracted.

Question 2:

My implementation was made with React as the frontend library.

To run this application please ensure you are in the root folder of the project (containing the package.json file) and run the command "yarn run start". This will start a localhost server and will open the web browser to the page. This application contains two pages, one is the contact list which can be searched for any key word, and secondly if a contact is pressed a page with more information is shown.