CardMakerSystem
Team: Republic
Developers: Milap Patel, Jason King, Huayi Zhang, Kevin Fortier
URL: https://card-storage.s3.us-east-2.amazonaws.com/cms_frontend/cms_frontend/index.html

Functionality:

1) Get all cards stored in database.
2) Create a new Card and add it to the database.
3) Delete a card stored in database.

Instructions:

1) **Get cards-** All cards are retrieved from database upon loading the card_maker_system. The request response cycle takes several seconds to complete. A successful request response cycle with result in the html table-element on the card_maker_system homepage being populated with the stored card data.
2) **Create Card-** A card is created by typing a value into the “Recipient” and “Event” fields as well as selecting “portrait” or “landscape” in the form-element at the top of the card_maker_system homepage and clicking the “Create Card” button. Newly created cards will be appended automatically to the table-element on the homepage.
3) **Delete Card-** A card can be deleted by checking the card’s radio button element in the card’s row in the homepage’s table-element followed by clicking the “Delete Card” button just above the table-element. In order to see that the card has been deleted, press the “Refresh Cards” button and wait for request-response cycle to complete. You will observe that the card you deleted is no longer listed in the table element.

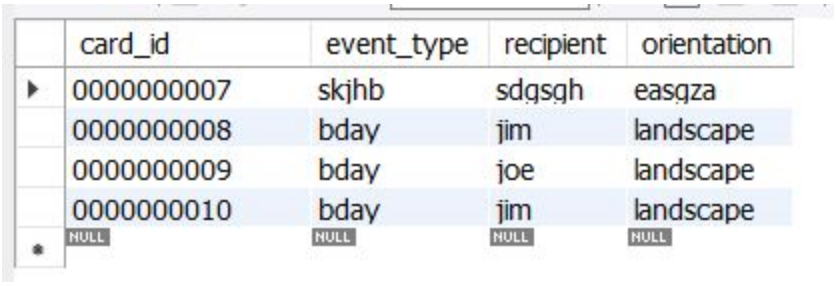
Database:

We have two tables in our database. The first table manages all the cards metadata such as event, recipient and orientation of a card. The id field for each card in this table is a unique key which is auto incremented by the database. The second table is for storing element metadata for each card. This table has the following attributes:

Element id -> unique key which is auto incremented by data base
Card id -> to link the element to a card in the card table
Page type -> to identify the page the element is going to be on in the card
Element type -> either image to text element
Text message -> if text element, this is where the message will be stored else it will be null for image
Text font -> same as above but stores the font type for the text (null if image)
Image source -> link to the image in S3 bucket used for rendering (null for text element)
X coordinate -> element’s x coordinate on the page
X coordinate -> element’s y coordinate on the page
Height -> height of the element
Width -> width of the element

To display the cards on initial landing page, the first table can be used to retrieve all the cards. Once a card is selected, all the elements can be retrieved from the second table given a card id. An element can also be updated or deleted using the second table given an element id.

Image of the first table:



card_id	event_type	recipient	orientation
0000000007	skjhb	sdgsgh	easgza
0000000008	bday	jim	landscape
0000000009	bday	joe	landscape
0000000010	bday	jim	landscape
NULL	NULL	NULL	NULL

Image of the second table:

element_id	card_id	page_type	element_type	text_message	text_font	img_src	x_coord	y_coord	height	width
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL