

# Node Dealer

Using the MEAN stack, create a single-page web app that shuffles a standard deck of 52 cards, and then deals them in a matrix of 13 columns by 4 rows. Calculate the percentage that the deck is in the correct order and store the deck configuration and the percentage in the database. Display an average percentage across all historical percentages. Display the deck on the screen with a button that is labeled "Deal Again" to deal the next deck. All business logic should be handled on the back end with node. Isolate your card generation, shuffling algorithm, and percentage calculations, as I will be reviewing that code. The site should be self-contained with all node modules and code zipped up. See below for details:

For every card that is in the correct suited row, add one point  
For every card that is in the correct value column, add one point

Max points = 104

	A	2	3	4	5	6	7	8	9	T	J	Q	K
(S)	[A]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[T]	[J]	[Q]	[K]
(D)	[A]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[T]	[J]	[Q]	[K]
(H)	[A]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[T]	[J]	[Q]	[K]
(C)	[A]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[T]	[J]	[Q]	[K]

This configuration would give you 104 points = 100% correct order (will probably never happen)

MongoDB: use mongo as the persistent datastore to save the configurations and percentages, give it a sequence number also – use mongoose.js schema for the data access layer

Express.js: use express as the web server to handle routing the calls from the front end

Angular.js: use angular framework as the front-end templating engine

Node.js: use node to handle all the back-end business logic and database access

Bonus: add a UI to access and display historical decks

Extra Bonus: highlight adjacent pairs, triplets, and quads (horizontal, vertical, or diagonal)