# HEADSTORM CHALLENGES

Front End Challenge:

* Company name title

For this I kept an google image of the HEADSTORM title in header bar of the webpage.

* Company logo as icon in browser tab

I included the logo of HEADSTORM from HEADSTORM uploads as a favicon for the webpage in browser tab.

* Contact us web form that captures contact information

I created a webpage in HTML headstorm\_form.html which takes contact information. I used BOOTSTRAP to ease my designing. I am also using JQuery to validate the information and displaying alerts indicating that all the fields are mandatory. I wrote all my validation part in a separate headstorm\_validation.js file.

* Google reCaptcha V3 implement in page. Submission of form requires Google captcha pass.

For this part I included a headstorm\_submit.php for server-side integration which on POST will check if the enduser is a spammer or not.

* Dump all the information from the form submission to browser console.

On a successful submit operation i.e., after validation it logs the contact information in browaser console and waits instead of reloading itself.

Back End Challenge:

* I created a JAVA EE project in Eclipse and designed GET and POST endpoints. For the 500 numbers instead of storing it in database I am storing them in-memory using List<Integer> variable. I am also returning the results in JSON format using GSON package. I extracted the project as a WAR file and dumped into TOMCAT server. I tested by giving input using “Advanced REST client” Google chrome extension.

Database Challenge:

* Part 1: Please review the current data stored in the NoSQL database and submit a new relational data model visualization.

I created a STARTUP\_DB.sql file which creates a relational database with all the attributes that has all the fields of given NoSQL database.

* Part 2: The next step is to create a solution for the client that can migrate the data from their current database to your new relational database.

For this I wrote a stored procedure which will take care of the inserting the records in the database. I created a Transcation\_Record class which has all the required attributes of our database as class members. I am taking the JSON file and first converting each record into my Transcation\_Record class object and then sending this object to stored procedure call which will insert the object’s class members as a record. In the end after inserting all the records I am displaying the number of rows that are inserted into the database.