* Introduce ourselves
* Our project was a fullstack clone, so users would posts questions and users would answer questions. Users can vote on answers
* User could sign up to the website and log in
* Website had a game aspect to it where it would keep track of their rank
  + Rank is based on the number of questions and answers the user would post
  + Higher ranking users got a nice color to their username
* We used html and css and javascript and json and jquerey fro the front end, and java and mysql for the backend.
* For the database we used XAMPP
* Lets go over a bit of the architecture. When a user makes a post, the front end will grab that information and send it as a json object to the Web API, which turns it into a java object. It then makes a call to the internal api, which then communicates with the business logic layer. The object is given to DAO layer, which then makes a query to create a new question. If there is a problem with the database, the DAO layer throws an exception which the business logic catches. Business logic handles that error appropriately and sends back a null value or a message.
* Demo how communication between the logic, dao and database work.
* Unfortunelty we were unable to get all the different parts communicating with each other properly but each part does work, and logic dao and database can talk, and the javascript can talk to the web api.