1. Document what, if anything, you would do differently if this were a production application and not an assessment? What tech would you use? How would you handle things differently if it needed to handle more users, more data, etc?
    a. I would use mysql instead of sqlite
    b. I would not have the logon screen and logoff interface the way it is (currently you type in an incorrect name and password to logout)
    c. I would definitely implement some framework so that it doesn't look ugly
    d. For more users I would create the ability for anyone to create a user, currently only an admin can create a user this is a huge time waste
    e. When a user logged in it would send them directly to their info instead of manually having to go there
    f. Its still in debug mode so that would have to change
    g. I worked from ticket to ticket this created some weirdness in the way I created the database. I would look at it more holistically and try drawing a database diagram.
    h. Passwords would be hashed
    i. The tech stack would be basically the same, and even though this has problems with some tweaks in the views it could scale relatively well.
2. Explain the implications of switching the type of data being used for mood values after an initial version is deployed. For example, if the value started as a float or a string, but needed to change to only support integers, how would you handle this change?
    a. There are several ways to handle this. Depending on what was needed. If the database needed to be uniform for some reason, then logging into the sql shell and programmatically switching them to some default may be the best way.
    b. Another node could be added to represent the new data giving the old data a default type for this new node and letting all nodes after have to have a value for this node
    c. The implications are mostly typing its not a terribly difficult thing to change in the sql shell, the problem is making sure your code now works with the new data type. And if it doesn't then it displays some feedback and informs the database manager that they missed one of the nodes.