Create a web-based note application (using any JS-MVC framework) that allows users to log in, see posted notes, and post new notes. Do NOT build registration, forgot password, etc. Pre-populate the application with 3 user accounts, Alice, Bob, and Charlie (use any database; you will NOT be evaluated on the basis of the database used). The result should be presentable (use any UI framework).

Requirements:

1. Users should be able to create notes that should be viewable only by them, and users should be able to delete, update, and modify each note.
2. Each note must be sent and reterieved in the JSON format (use the REST backend or in-browser (HTML5 localstorage)).
3. Notes should be shown in chronological order. Show notes in the form of cards. Not more than 4 notes should be displayed at any instance (design the UI such that up to 4 notes fit in a desktop screen).
4. User should be able to mark each note with priority (in a range of 1 (Highest) to 3 (Lowest)).
5. Users should be able to create, delete, and update on the same page or same view.
6. Restrict users to a maximum note size of 50 words.
7. User should able to delete multiple notes on a single click.
8. Different background colors should be assigned to each note according to its priority (e.g. 1-Red, 3-Green etc.)

Bonus points:

1. Write a script to pre-populate the database with notes containing 50 random characters. Each user should have 30 notes each.
2. Use the lazy-load technique to display notes as the user scrolls down.

Note that you are not expected to write each snippet of code from scratch. Avoid making the application more complicated than it needs to be; the purpose is to build the application as if it were a real-life projects and not to show us which all libraries and frameworks you know. You should send us code, an SQL file to initialize the database (if needed), and any other instructions on how to set up and run your application.

You will be evaluated on the following factors:

1. Does the application provide all the functionalities specified?
2. Is the code simple, minimal, readable, organized well, and easy to maintain?
3. Is the UI free from serious aesthetic or UX-related issues?
4. Time taken to build the application

Finally, if you have any doubt regarding the requirements, please do NOT ask us. Make a reasonable assumption and proceed. The ability to make sensible assumptions given partial (or even self-contradictory) instructions is an important part of the job!