

Assignment 2 – HTML, CSS, JS, React and MongoDB

Software System Development – Monsoon 2023

Due Date: 3rd October, 2023, 11.59 pm

Instructions:

- This assignment is an individual submission.
- Total Marks of 100 Marks with duration of 3 weeks.
- All script submissions should be submitted via moodle
- Inputs/output should fit the criteria mentioned in the respective question.
- All other conditions are open to your interpretations.
- Evaluation will be conducted based on a fixed grading rubric (syntax, logic, input and output) and the marks are divided as per prescribed weightage in respective questions.
- For queries, reach out to TAs via Moodle

Submission Criteria:

- Please create one ZIP file as <rollnumber>_A2.ZIP
- Include all the files of the respective questions in their respective directories. Follow the directory structure given below.
- Each question will have three folders (html,css,js).Required files should be in these folders only (For e.g. html file in html folder , css in css one)
- For Example:

2022201079_A1.zip

```
|__2022201079_A2_Q1
    |__html,css,js folder
|__2022201079_A2_Q2
    |__html,css,js folder
|__2022201079_A1_Q3
    |__html,css,js folder
|__README.md
```

Question 1: (30 Marks)

Redesign the page (Link : <https://sites.google.com/site/charusharmacseiith/>)

This question is open to your interpretation and Designing ideas.

Important points to Remember:

1. Refer the content from the link, Keep the information as it is
2. **All links should be intact**

(Note : allowed tech stack is HTML,CSS, Javascript,bootstrap,react.)

Question 2: (30 Marks)

We all like music. What if we integrate it with our SSD Learnings and make something of our own!

The task is to play with **spotify APIs** and show content on a webpage of your own.

For-eg: new release API of spotify is to get new releases from its platform.

(Link: <https://developer.spotify.com/documentation/web-api>)

This question is open to your experimentation with API and web page design

Expected result: Using spotify APIs to show content on a webpage.

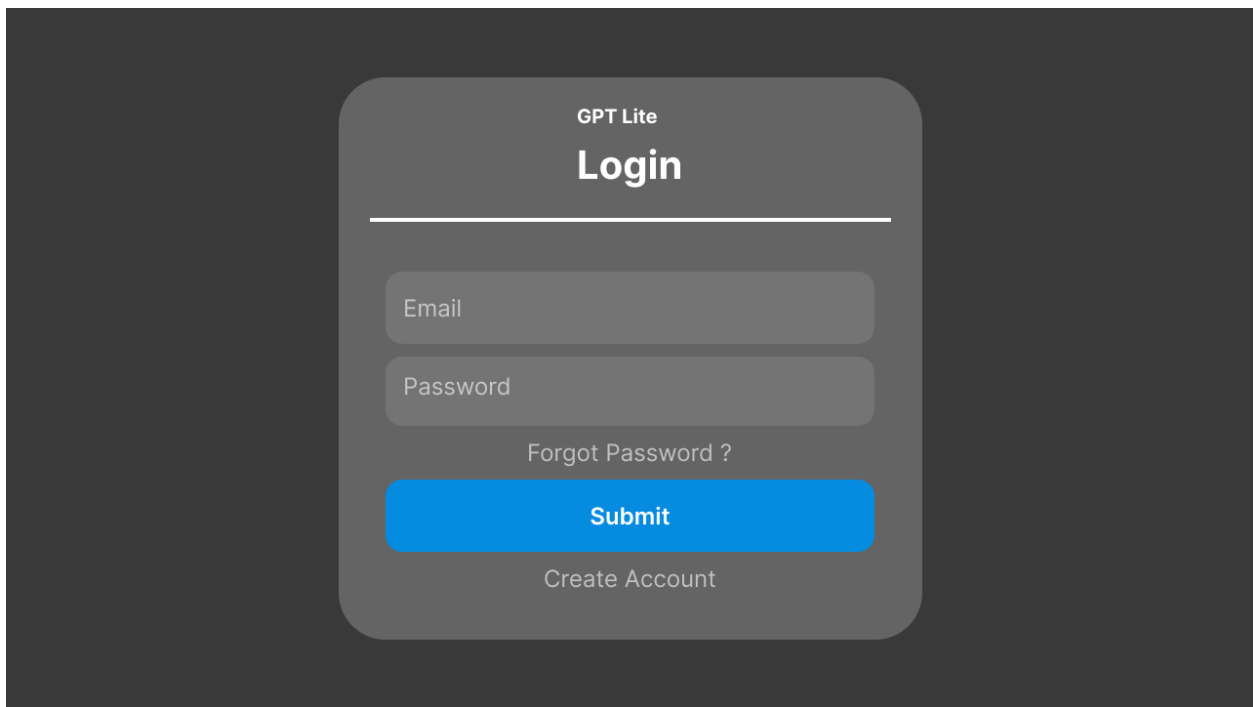
Question 3: (40 Marks)

Your task is to create a working Lite version of **Chat GPT** using **ReactJS** and **MongoDB**. The following are the points you need to implement.

1. Login Page **(10 Marks)**
2. Register Page **(10 Marks)**
3. Chat Area **(10 Marks)**
4. Admin Page **(10 Marks)**

Login Page:

It shows email and password inputs, which is connected to MongoDB where each email ID should be unique. (Remember that there are 2 types of users: one is **normal user** and other is **admin**)

The image shows a login page for 'GPT Lite'. It features a dark gray background with a central light gray rounded rectangle. At the top of this rectangle, 'GPT Lite' is written in small white text, followed by 'Login' in a larger white font. A horizontal white line separates the header from the input fields. There are two input fields: 'Email' and 'Password', both with light gray borders and rounded corners. Below the 'Password' field is a link that says 'Forgot Password ?' in white text. A prominent blue button with the word 'Submit' in white is located below the links. At the very bottom of the light gray rectangle is a link that says 'Create Account' in white text.

Register Page:Login Page:

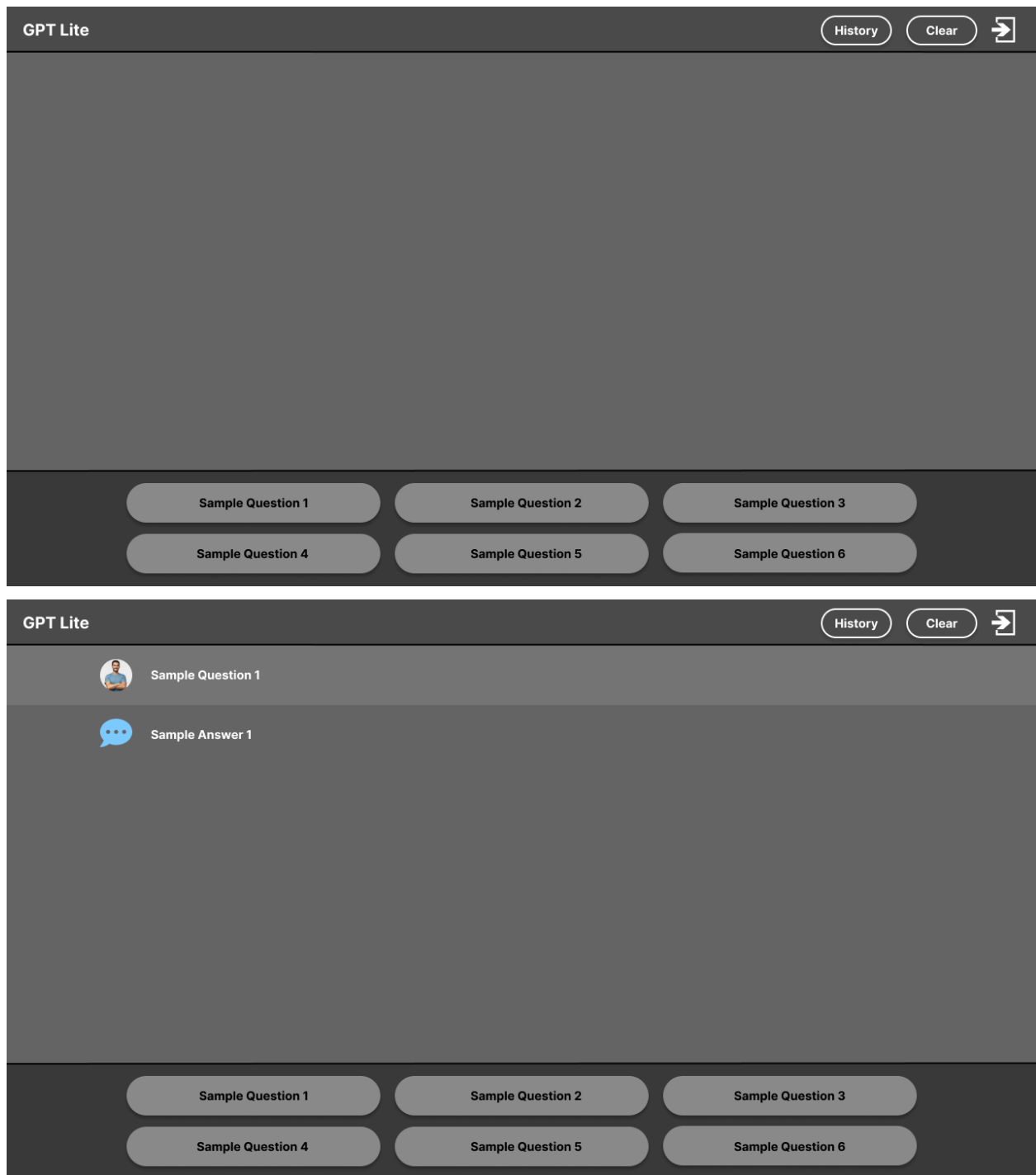
It shows email and password inputs, which is connected to MongoDB where each email ID should be unique. (Remember that there are 2 types of users: one is normal user and other is admin)

It will take **Name,Role(User/Admin), Email, and Password** from the user to register. (Note: Check for the email if it already exists in the DB) (Reference: Similar to Login page above)

Chat Area:

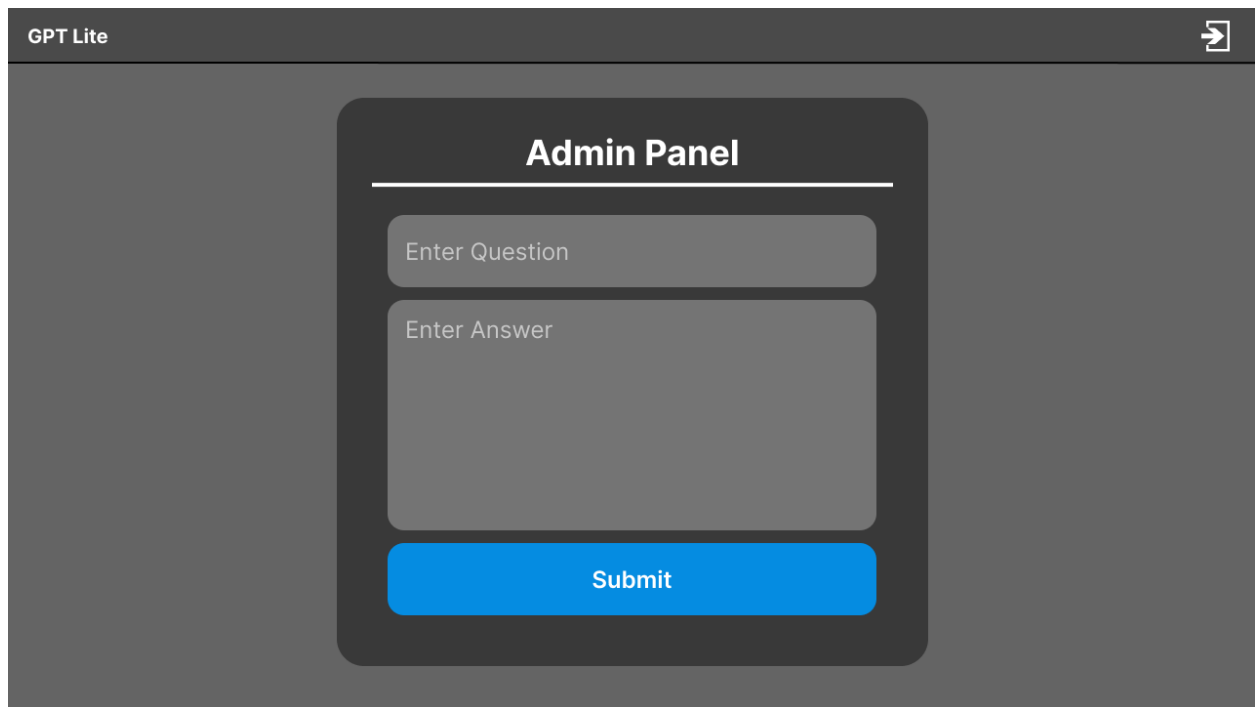
The following things you need to implement in this area:

1. You must save the **history** of the chat.
2. Logout option
3. Clear conversation (erase all history)
4. All question in **Drop Down menu** for user to ask (fetched from DB)
5. When a user clicks on any question the answer for that respective question should come on the screen.
6. History Button which will show previously asked questions. (Either in new page or as popup window)



Admin Page:

This page is **only for admins**. Admins can add new questions to the database or can change any question or its answer.



Note:

1. You can use Bootstrap for designing purposes
2. Passwords should not exist in plaintext in the database.
3. Without login, no page accessible, redirection to login page required.
4. Above screenshots are just an example. You are free to design according to you.

Bonus Section (Optional)

1. Except for the login & Register page, give the choice for light mode and dark mode.
2. Profile page in navbar which will have the option to change user details like name,email, etc.