perplexity

list all the possibel questions that can be asked based on the above questions include already asked ones as well

Here is a comprehensive, categorized list of all possible questions that can be asked in the MAD viva (across Levels 1, 2, and 3), including those that have already been asked, based on your extensive history and the trends observed:

General Project and Self-Assessment

- Introduce yourself and your background.
- Give an overview or walkthrough of your project.
- Why did you choose this project topic?
- What are the main functionalities of your app?
- What would you improve about your project?
- Rate your app (out of 5 or 10) and justify the rating.
- Did you develop the project yourself? How much external help or AI did you use?
- What challenges did you face, and how did you solve them?

Frontend (Vue.js, HTML/CSS, JS)

- What is Vue.js? Why did you choose it over React or Angular?
- What are Vue lifecycle hooks? Explain each (`created`, `mounted`, `updated`, `destroyed`, `watch`).
- What is the difference between `v-if` and `v-show`?
- How is `v-bind` different from `v-model`? What is two-way binding?
- Explain `$router` vs `$route`. How is dynamic routing handled?
- How do you implement a route link and programmatic navigation (e.g., `this.$router.push('/url')`)?
- What is a Vue component? How do you create and reuse components?
- What does the `class="{ ... }"` binding do?
- How is data passed between parent and child components? (props, emit)
- What is the difference between single-page applications (SPA) and traditional websites?
- How do you handle form validation (e.g., only numbers in pincode fields)?
- How do you demonstrate conditional rendering or dynamic lists in Vue?
- What is the use of `watch` in Vue?

- How do you fetch data from the backend and display it in Vue?
- Explain the difference between `const`, `var`, and `let` in JavaScript.
- How do you style components (CSS selector types: class vs. id, inline/internal/external CSS)?
- How do you make your frontend responsive (CSS, Bootstrap, Flex, Grid)?
- How does the virtual DOM work in Vue?
- What are the types of ways to store data in the browser? (localStorage, sessionStorage, cookies)
- What is the difference between `fetch` and `axios`?

Backend (Flask/Python & API)

- What is Flask? Why did you choose it?
- What are blueprints in Flask?
- What does `config.py` do? What are secret tokens, backend_url, and result_backend_url?
- Explain the authentication and authorization process in your project.
- What are Flask decorators (`@app.route`, `@roles_required`, `@jwt_required`, `@shared_task`) and their purpose?
- Describe the flow of user login/authentication in your app.
- What is the difference between authentication and authorization? How is RBAC (Role-Based Access Control) implemented?
- How do you handle environment variables and configuration?
- What is CORS? How is it handled?
- What are CSRF attacks, and how can they be prevented?

Database and ORM (SQLAlchemy/Models)

- What is an ORM? What is SQLAlchemy?
- Explain your `models.py`. Why did you define the columns/relationships as you did?
- Explain one-to-one, one-to-many, and many-to-one relationships with examples from your project.
- How do you enforce uniqueness or constraints (e.g., no duplicate campaign names)?
- What is a primary key? How is it different from a unique key?
- How do you perform database migrations?
- How would you change your app to use PostgreSQL (or any other DB)?
- What is cascading delete? How is it implemented in SQLAlchemy?
- What is the difference between JOIN, LEFT JOIN, and how do you use aggregate functions?
- Explain the difference between LIKE vs. ILIKE in SQLAlchemy queries.
- How would you handle errors if someone tries to insert a string into an integer field?

- How do you handle foreign keys?

APIs, REST, and Web Technologies

- What is an API? What is a REST API?
- Explain common HTTP methods (GET vs POST vs PUT vs PATCH). When do you use each?
- What are status codes 200, 400, 401, 403, 404, 500?
- How is data sent and received between frontend and backend (JSON, URL params, Form data)?
- What are webhooks? How are they different from APIs and WebSockets?
- What is Server-Sent Events (SSE)?
- What is the difference between $router and $route?

Asynchronous Tasks & Caching (Celery, Redis)

- What is Celery? What are its components (worker, beat, broker)?
- How do you configure and use Celery in your project?
- How does Celery beat work? How do you schedule tasks or change intervals?
- What is Redis? What is its primary function (broker, cache, result-backend)?
- How is caching implemented in your project? Show caching in routes.
- What is the difference between Redis and Memcached?
- How do you demonstrate emails being sent (SMTP setup, Celery task)?
- How do you clear or update cache after data changes?

Practical Coding, Debugging, and Modifications

- Change background or a UI color (login box to yellow, homepage to red, etc.).
- Move/modify layout elements (navbar to bottom, center a button, adjust font size/color).
- Add a new functionality/feature (search, show alerts, add button, dynamic chart).
- Show only numeric input (for pincode), validate password length, constrain duplicate entries.
- Fetch and display specific data (e.g., show usernames for a given spot ID).
- Implement or adjust live routing (`this.$router.push()`).
- Make an inline logout function.
- Modify database models to add/remove a field, enforce relationships or constraints.
- Change a Celery Beat schedule or demonstrate a scheduled task.
- Demonstrate caching (e.g., show difference before and after cache).
- Show emails being sent (from a celery task or route).

- Fix a logical error (e.g., venue should not be deleted if bookings exist, prevent accidental deletion).

Security and Best Practices

- How do you store and secure JWT tokens (why localStorage vs. sessionStorage)?
- How do you protect sensitive endpoints?
- What is the structure of a JWT, and how do you verify or decode it?
- What security measures have you implemented (prevent SQL injection, secure REST endpoints, secure cookies, etc.)?
- How do you prevent showing sensitive data in the dev tools/network tab?

Version Control (Git)

- How do you use Git in your project?
- What is the purpose of branching?
- How do you resolve merge conflicts?

Miscellaneous/Other

- What is the difference between a module and a component?
- What is SSE (Server-Sent Events)?
- What is the use/purpose of `cli.py`?
- What is the purpose of `.env` files, and what is stored there?
- What is the difference between inline, internal, and external CSS?
- What is a decorator in Python, and why/where do you use it?
- Explain the difference between session, localStorage, and cookies.
- What is a SPA (Single Page Application)?

This list combines all historical and possible questions based on observed patterns, past experiences, and documented viva sessions. Use this as a master list for thorough preparation—review and be confident explaining every point, practicing code changes where necessary, and focusing on understanding rather than rote memorization.