Backend (Flask) + Frontend (Vue.js) Viva Questions

Q: What is Vue?

A: Vue.js is a JavaScript framework used to build user interfaces and single-page applications (SPAs).

Example: In a Vue to-do app, you can dynamically add tasks and update the list instantly without reloading the page.

Code:

<input v-model='task'> binds the input to your Vue data.

Q: What is a reactive framework?

A: A reactive framework automatically updates the UI when the data changes.

Example: In Vue, changing the value of a variable like `count` will instantly update the HTML showing `{{ count }}`.

Q: What is memoization?

A: Memoization is an optimization technique where the results of expensive function calls are cached.

Example (Python):

@lru_cache

def fib(n): return n if n <= 1 else fib(n-1) + fib(n-2)

Q: What is caching?

A: Caching stores frequently accessed data so future access is faster.

Example: Flask can use Redis to cache API responses and reduce database hits.

Q: Difference between memoization and caching?

A: Memoization caches function results; caching can store any reusable data like files, API results, etc.

Example: Memoization is used inside functions, caching might be used at the server or browser level.

Q: What is Celery?

A: Celery is a Python library to run background tasks asynchronously.

Example: After a user signs up on your Flask app, Celery can send a welcome email in the background.

Q: What is Redis?

A: Redis is an in-memory data store, often used for caching and message brokering.

Example: Flask can use Redis to store session data or task queues for Celery.

Q: Difference between Memcached and Redis?

A: Redis supports more data types and persistence; Memcached is simpler but only supports key-value storage.

Example: Use Redis for sessions and lists; Memcached for lightweight caching.

Q: What is ORM?

A: ORM (Object-Relational Mapping) lets you interact with the database using objects instead of SQL.

Example: `User.query.all()` in SQLAlchemy instead of `SELECT * FROM users;`

Q: What is SQLAlchemy?

A: SQLAlchemy is a Python ORM for managing databases using models and queries in Python.

Example: Define a User class as a table, and use Python to query/update rows.

Q: What is Flask?

A: Flask is a lightweight Python web framework for building web applications and APIs.

Example: You can create routes like `/api/users` that return JSON data for your frontend.

Q: What is an API?

A: API (Application Programming Interface) allows software to communicate.

Example: Your Vue frontend fetches data from a Flask backend API endpoint `/api/data`.

Q: What is a webhook?

A: A webhook lets external services notify your app when an event happens.

Example: Stripe sends a webhook to your Flask app when a payment is successful.

Q: What is AJAX?

A: AJAX lets you fetch or send data to the server without reloading the page.

Example: Vue uses `fetch('/api/data')` or Axios to make async requests to Flask.

Q: What are Vue lifecycle hooks?

A: These are special Vue methods that run during a component's life (creation, mounting, updating, destroying).

Example: Use `mounted()` to fetch data from your Flask API when the component is loaded.

Q: Difference between v-if and v-show in Vue?

A: `v-if` adds/removes elements from the DOM; `v-show` only toggles visibility using CSS.

Example: Use `v-if` for conditional rendering and `v-show` for toggling UI frequently.

Q: What are slots in Vue?

A: Slots let you pass HTML content from parent to child components.

Example:

<MyCard><template v-slot:header>Title</template></MyCard>

Q: What is v-model in Vue?

A: `v-model` creates two-way binding between input elements and Vue data.

Example: <input v-model='name'> automatically syncs the `name` variable with the input.

Q: What is v-bind in Vue?

A: `v-bind` binds HTML attributes to dynamic data.

Example: <img v-bind:src='imageUrl'> sets the image source based on Vue data.

Q: What is a component in Vue?

A: A component is a reusable block of UI in Vue, like a button or form section.

Example: A LoginForm component can be reused across different pages.

Q: What is a Single Page Application (SPA)?

A: An SPA loads a single HTML page and updates dynamically without reloading.

Example: Your Vue app updates views using Vue Router without reloading the page.

Q: Difference between authentication and authorization?

A: Authentication verifies identity (login); authorization checks access permissions (roles).

Example: Only authenticated users can access `/dashboard`; only admins can access `/admin`.

Q: What is RBAC?

A: RBAC (Role-Based Access Control) limits access based on user roles.

Example: Flask with JWT can allow only admins to delete users.

Q: What is JWT?

A: JWT (JSON Web Token) is used for securely transmitting user info between client and server.

Example: After login, the server sends a JWT to Vue, which is attached to future API calls.

Q: What is CORS?

A: CORS (Cross-Origin Resource Sharing) determines which domains can access your API.

Example: Allow your Vue frontend at `localhost:8080` to call your Flask backend at `localhost:5000`.

Q: What is CSRF?

A: CSRF (Cross-Site Request Forgery) is an attack where malicious websites perform actions on behalf of a user.

Example: Protect Flask forms with CSRF tokens to avoid these attacks.

Q: What is template inheritance?

A: Template inheritance allows reusing common HTML structure using base templates.

Example (Flask + Jinja): base.html has layout; child.html fills in content blocks.

Q: Difference between PUT, PATCH, POST, GET?

A: GET: Read data

POST: Create new data

PUT: Replace existing data

PATCH: Update part of existing data.

Example: Use POST to register users; PATCH to update email only.

Q: What is lazy loading?

A: Lazy loading loads content only when needed.

Example: Vue only loads components or images when they're visible, speeding up page load.

Q: What is a decorator in Python?

A: A decorator modifies behavior of functions or routes in Flask.

Example: @login_required decorator checks user login before allowing access.

Q: What is scaling (vertical/horizontal)?

A: Scaling improves performance by increasing resources.

Vertical: Upgrade server

Horizontal: Add servers

Example: Use load balancers and add more Flask instances for horizontal scaling.

Q: Difference between local storage, session storage, and cookies?

A: LocalStorage: Persistent storage

SessionStorage: Cleared on tab close

Cookies: Sent with every request

Example: Store JWT token in localStorage to persist login.

Q: What is SQL Injection?

A: SQL Injection is an attack where malicious SQL is inserted into queries.

Example: Always use parameterized queries in Flask SQLAlchemy to prevent this.

Q: What is a Man-in-the-Middle (MITM) attack?

A: MITM is where an attacker intercepts communication between client and server.

Example: Use HTTPS and TLS to prevent MITM attacks.

Q: What is an aggregate function in SQL?

A: Aggregate functions perform calculations on data sets (e.g., COUNT, SUM).

Example: SELECT COUNT(*) FROM users;

Q: What is a JOIN in SQL?

A: JOIN combines rows from two or more tables using related keys.

Example: SELECT * FROM users JOIN orders ON users.id = orders.user_id;

Q: What is a foreign key, primary key, unique key?

A: Primary key: Unique row identifier

Foreign key: Reference to another table's PK

Unique key: All values must be unique

Example: orders.user_id is a foreign key referencing users.id.

Q: What is a message queue/task queue?

A: Used to handle background tasks in order.

Example: Flask + Celery + Redis queue for sending emails, processing reports.

Q: What is a callback function?

A: A function passed into another to be called later (common in JS).

Example: setTimeout(() => alert('Hi'), 1000); // arrow function is a callback.

Q: What is a virtual environment?

A: A virtual environment isolates project dependencies in Python.

Example:

```
python -m venv venv
source venv/bin/activate
```