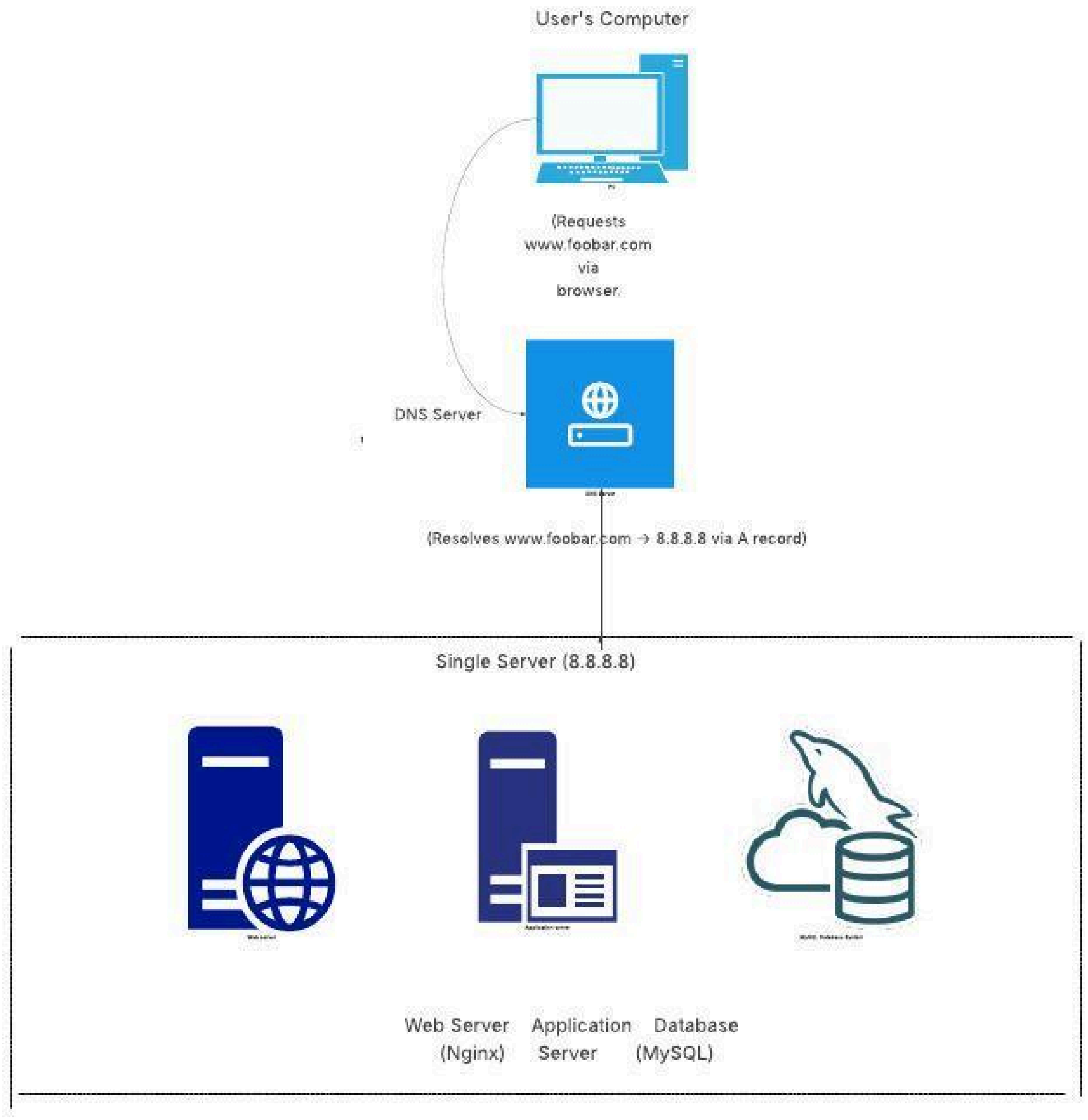
Diagram (ASCII Representation)



Component Explanations

1. Server:
   A physical or virtual machine that provides services (e.g., hosting a website). In this case, it runs Nginx, an application server, and MySQL.
2. Domain Name (foobar.com):
   A human-readable address that maps to the server’s IP (`8.8.8.8`) via DNS. Users type `www.foobar.com` instead of the IP.

3. DNS Record in `www.foobar.com`:
   An A record maps the domain name to the server's IPv4 address (`8.8.8.8`).
4. Web Server (Nginx):
   Handles HTTP requests:
   - Serves static files (HTML, CSS, JS).
   - Routes dynamic requests to the application server.
5. Application Server:
   Executes the codebase (e.g., Python/Node.js/PHP app) to generate dynamic content (e.g., user profiles, transactions).
6. Database (MySQL):
   Stores and manages structured data (e.g., user accounts, posts). The application server queries it to fetch/update data.
7. Server-User Communication:
   - User enters `www.foobar.com` → DNS resolves it to `8.8.8.8`.
   - Browser sends HTTP request to the server.
   - Nginx processes the request: serves static files or proxies to the app server.
   - App server generates a response (using MySQL if needed) → returned to the user.

Limitations

1. Single Point of Failure (SPOF):
   If the server crashes, the entire website goes down. No redundancy.
2. Downtime During Maintenance:
   Deploying new code or restarting services requires downtime, making the site temporarily unavailable.
3. Cannot Scale Under High Traffic:
   - One server has limited CPU/RAM/bandwidth.
   - Database and app server compete for resources, leading to slowdowns or crashes under load.