IITGN

CS 331 - CN

DNS RESOLVER

FINAL PROJECT PRESENTATION

OUR TEAM

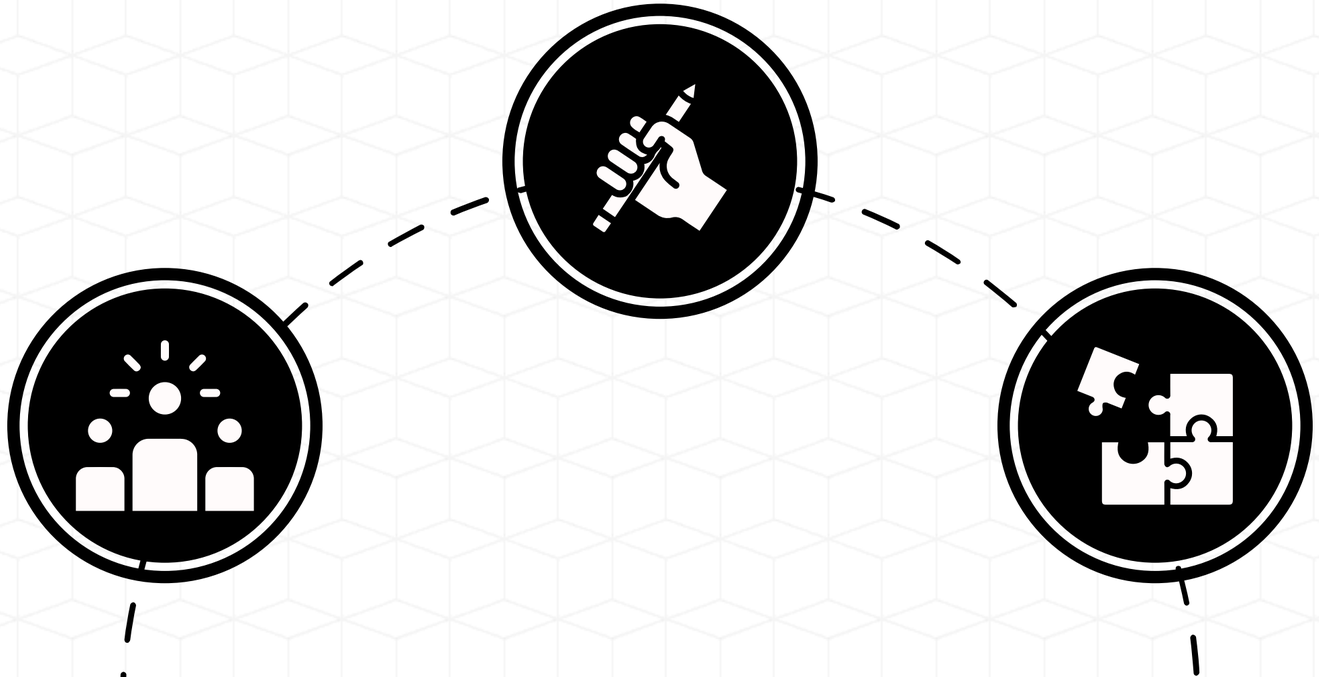
- *Chirag Patel - 22110183*
- *Ruchit Jagodara - 22110102*

LET'S DEMOSTRATE

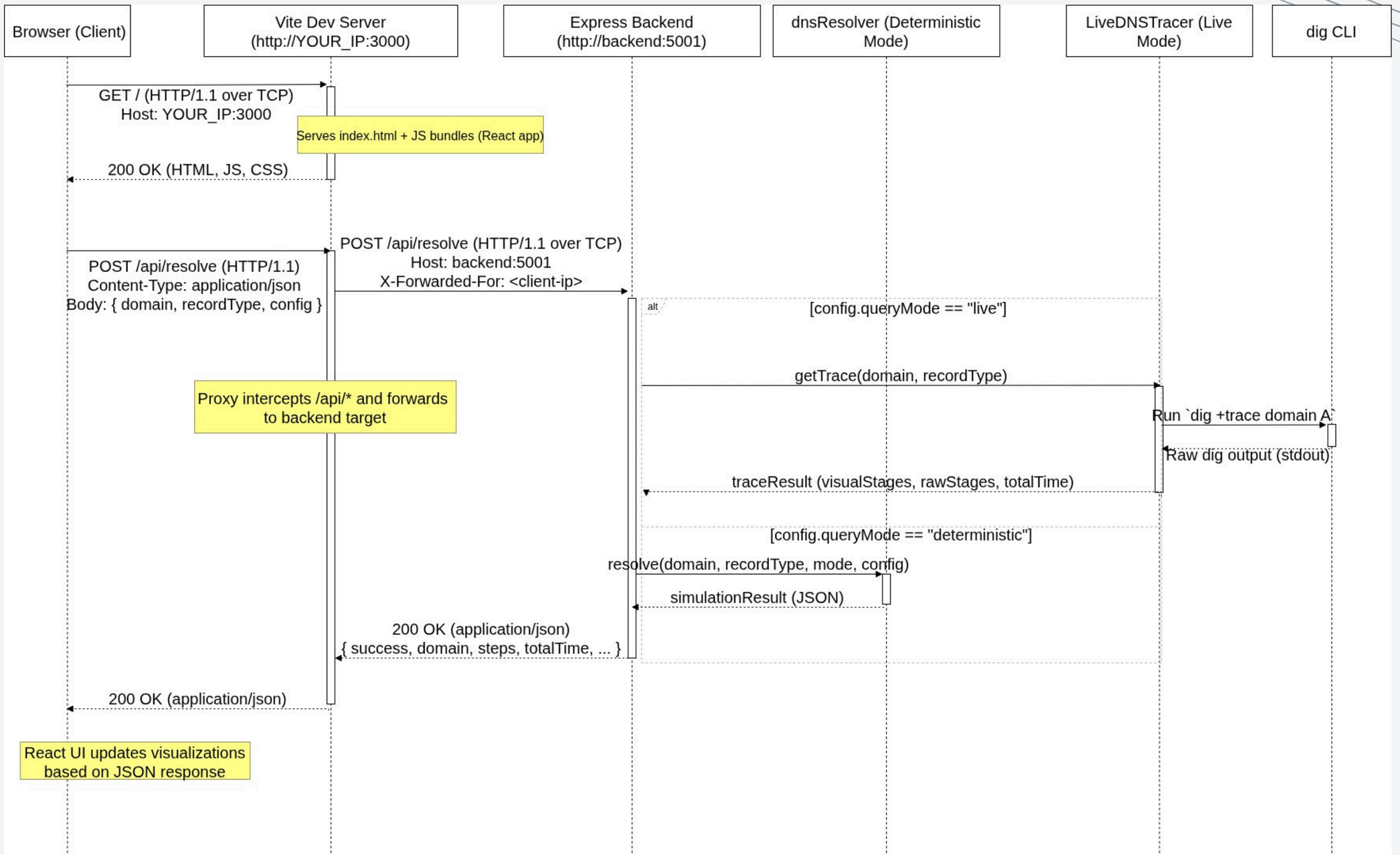
DNS resolver demo LINK

<<< FEATURES >>>

- Live Simulation mode
- Deterministic Simulation mode
- Attack Scenarios
- Security Protocols (DoT, DoH, DNSSEC)



OVERALL ARCHITECTURE



SIMULATOR - LIVE MODE

- Executes a real DNS resolution workflow for a given domain and record type using the system’s DNS tools.
- Produces a faithful, stage-by-stage account of the query path (root → TLD → delegated zones → authoritative), including retries, failures, timing, and DNSSEC-related records encountered in the trace.
- Feeds two UI surfaces: a step-level animated visualization and a results/timeline panel that exposes live-specific data such as transport attempts, DNSSEC records seen at each hop, and the raw dig output for verification.

ALTERNATIVES CONSIDERED

1. dig +trace command

Description:

The dig +trace command performs a complete iterative resolution starting from the root servers and following delegations down through the TLD, domain, and finally to the authoritative nameservers.
It automatically issues multiple DNS queries internally, printing all delegation steps in the correct sequence.

Advantages:

1. Fully automatic
2. Handles DNSSEC, EDNS(0), TCP fallback
3. Provides delegation hierarchy and server IPs
4. Simple and widely available

Disadvantages:

1. Semi-structured output
2. Limited customizations hence less control

ALTERNATIVES CONSIDERED

2. Implementing Direct DNS Protocol Queries

Description:

We can query servers directly using a DNS library such as dnspython in Python.
By disabling recursion (RD=0) and following referrals ourselves, we can recreate the full iterative process programmatically:

Advantages:

1. Complete Control over the process
2. structured JSON traces

Disadvantages:

1. Rate-limited if many direct queries are sent to root/TLD servers
2. Significant implementation effort and time

ALTERNATIVES CONSIDERED

Criterion	dig +trace	dig (manual)	dog (JSON)	q (JSON)	Direct Protocol (dnspython)
Ease of Use	5	3	4	4	2
Automation of Full Trace	5	2	1	1	2
Output Structure (Machine-Readable)	2	2	5	5	5
Protocol Control / Flexibility	2	4	3	3	5
DNSSEC / EDNS Support	5	4	2	2	4
Implementation Complexity	5	3	4	4	1
Reliability / Standard Compliance	5	4	3	3	3
Best for Educational Simulation	4	3	4	4	5
Overall Suitability (for our simulator)	5	3	3	3	4

ATTACK SCENARIOS

- Executes a real DNS resolution workflow for a given domain and record type using the system’s DNS tools.
- Produces a faithful, stage-by-stage account of the query path (root → TLD → delegated zones → authoritative), including retries, failures, timing, and DNSSEC-related records encountered in the trace.
- Feeds two UI surfaces: a step-level animated visualization and a results/timeline panel that exposes live-specific data such as transport attempts, DNSSEC records seen at each hop, and the raw dig output for verification.

SIMULATION - DETERMINISTIC MODE

- Simulates the full DNS resolution journey for a given domain and record type without performing real network lookups.
- Walks through the canonical path (client → resolver → root → TLD → authoritative), generating educational, step-level events that illustrate referrals, delegations, and the final answer.
- Offers configurable knobs for network latency and packet loss, optional failure scenarios (timeouts, NXDOMAIN, SERVFAIL), and conceptual DNSSEC validation cues including a toggle to force a validation failure.
- Produces a normalized results payload (steps, metrics, final answer, errors) that drives both the visualization and the timeline/results panels.

SECURITY PROTOCOLS

- Teaches how DNS security layers work by simulating three tracks: DNSSEC (chain of trust), DoT (DNS over TLS), and DoH (DNS over HTTPS).
- Visualizes the order of operations and actors involved: client, recursive resolver, upstream secure endpoint, and signing/validation steps for DNSSEC.
- Presents step-level animations and concise annotations to build conceptual clarity without exposing raw packet headers or performing real cryptographic handshakes.

IMPROVEMENTS

1. **Animations to include failures and fallbacks**
2. **Interactive Cache**
3. **Parameterized variant of Attack Scenario**

THANKS FOR YOUR TIME