

# UF-Xray Viva Guide

Version: 1.0 Date: 2025-09-20

## 1. Project Overview

- **Purpose:** Full-stack security helper with URL, File, and Log analysis, plus Security News.
- **Frontend:** React + Tailwind CSS, deployed on GitHub Pages.
- **Backend:** Node.js (Express) + Python scanners, deployable on Render via Docker.

## 2. React Basics Used Here

- **Functional Components + Hooks:** `useState` for UI state, `useEffect` for data fetching.
- **Routing:** `react-router-dom` with `HashRouter` for static hosting compatibility.
- **API Integration:** `src/utils/api.js` wraps `fetch`; base URL from `REACT_APP_API_URL`.
- **Styling:** Tailwind utility classes; final build uses fixed light theme.

## 3. Frontend Architecture

- **Navbar** (`src/Components/Navbar.js`): Responsive nav, mobile "Scan" is collapsible with auto-close.
- **Home** (`src/Components/Home.js`): Hero/shortcuts + News preview.
- **News** (`src/Components/News.js`): Lists headlines, routes `http://` images via backend proxy.
- **AnalyzeURL/File/Log:** Forms to send inputs to backend; shows results; supports downloads.
- **Entrypoint** (`src/index.js`): Forces light theme, normalizes URL for `HashRouter`.

### Data Flow Example: News

1. On mount or when limit changes, component fetches `/api/news?limit=...`
2. Sets `items` state; renders cards with title, summary, meta.
3. If image URL is `http://`, uses `/api/news-image?src=...` proxy to avoid mixed-content on HTTPS.

## 4. Tailwind CSS Embedding

- **Buttons:** `px-3 py-1.5 text-sm rounded bg-blue-600 text-white hover:bg-blue-700`
- **Cards:** `rounded-2xl border border-gray-200 bg-white shadow hover:shadow-lg`
- **Layouts:** `grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6`
- **Theme:** Removed dark-mode variants as per final requirement; fixed to light.

## 5. Backend Overview

- **Express App** (`server/server.js`): endpoints for scans, news, health check, and image proxy.
- **Python Integration**: `child_process.spawn` runs scripts and returns JSON results.
- **News Aggregation** (`server/news.js`): rss-parser over multiple feeds, 4h cache, image selection heuristics.
- **CORS**: Open for public access; allowed headers include Authorization.

## 6. Environment

- **REACT\_APP\_API\_URL**: production base URL for API; configured via GitHub Actions secret or `.env.production`.
- **Local Dev**: base URL falls back to `http://localhost:5000` when running on localhost.

## 7. Deployment Summary

- **Frontend**: GitHub Actions workflow (`.github/workflows/deploy.yml`), Node 18, `npm ci`, build, deploy Pages.
- **Backend**: Render via Docker (`server/Dockerfile`) and `render.yaml` blueprint.

## 8. Security & Reliability

- **Input validation** for URL scan; file cleanup after processing.
- **Error handling** around Python processes; health endpoint.
- **Image proxy** adds caching, redirects, and timeout handling.

## 9. Performance & UX

- **News limit** and quick refresh.
- **Mobile nav** collapsible Scan section reduces scroll.

## 10. Common Viva Questions (with answers)

- **Why HashRouter on GH Pages?** Static hosting cannot rewrite URLs; hashes keep routes in the client.
- **How do env vars reach React?** CRA inlines `REACT_APP_*` at build time; we read them in the API layer.
- **Why image proxy?** To prevent mixed-content blocking for `http://` images on an `https://` page.

- **How are PDFs/JSON produced?** Helpers in `src/utils/pdf.js` and `src/utils/download.js`.
- **How does Node run Python?** `spawn(PYTHON_BIN, [script, args])`, parse JSON from `stdout`.
- **Benefits of Tailwind?** Speed, consistency, responsive utilities without manual CSS.

## 11. Local Run

```
Backend:  cd server && npm install && npm start
Frontend: npm install && npm start  (open http://localhost:3000)
```

## 12. Troubleshooting Highlights

- **GH Pages 404:** Repo Public, Pages Source = GitHub Actions, wait 1–5 minutes, use `/#/` routes.
- **Prod API fails:** Set `REACT_APP_API_URL` secret, redeploy, verify `/healthz`.
- **CI warnings:** Build sets `CI=false` to avoid failing on warnings.