**Week 1 (9 June – 15 June): Planning, Architecture, and Infra Setup**

**Activities:**

* Finalize project scope, use-cases, and data sources (MISP, OTX, etc.)
* Design system architecture (frontend, backend, AI, APM)
* Define data ingestion strategy (real-time WebSocket, batch REST)
* Set up cloud infrastructure (AWS/GCP with Docker)
* Set up GitHub/GitLab repo with CI/CD pipeline

**Tools:**

* Jira, Draw.io/Lucidchart, GitHub Actions/GitLab CI
* Docker, AWS EC2/ECS or GCP Compute Engine
* REST, WebSockets, JSON

**Week 2 (16 June – 22 June): Backend, Ingestion, and APM Integration**

**Activities:**

* Implement backend APIs (FastAPI or Express)
* Integrate threat data ingestion (polling + WebSocket)
* Add secure authentication (OAuth 2.0, JWT)
* Begin custom APM integration (OpenTelemetry)
* Store structured/unstructured data (PostgreSQL + MongoDB)
* Start internal Grafana dashboards (APM metrics)

**Tools:**

* FastAPI / Node.js, PostgreSQL, MongoDB
* OpenTelemetry, Prometheus, Grafana
* OAuth 2.0, JWT

**Week 3 (23 June – 29 June): Frontend + AI Model Development**

**Activities:**

* Develop React.js dashboard (heatmaps, alerts, APM charts)
* Role-based dashboard views
* Build and test anomaly detection model (time-series or frequency based)
* Containerize AI model and deploy via FastAPI microservice

**Tools:**

* React + TypeScript, Tailwind CSS/Material UI
* D3.js / Recharts / Chart.js
* Scikit-learn, Pandas, XGBoost
* Kafka (optional) or simple scheduled cron jobs

**Week 4 (30 June – 7 July): Testing, Security, Final Deployment**

**Activities:**

* Perform functional testing (Postman), load testing (JMeter), and security testing (OWASP ZAP)
* Set up alerting (Prometheus Alertmanager / CloudWatch)
* Containerize and deploy complete stack (ECS/EKS or GKE)
* Final documentation and code freeze

**Tools:**

* Postman, JMeter, OWASP ZAP
* Docker Compose / Kubernetes (EKS/GKE)
* Swagger/OpenAPI, MkDocs

**Week 5 (8 July – 14 July): Buffer, Polishing & Handover**

**Activities:**

* Polish UI/UX and fix any bugs
* Conduct internal demo and final QA pass
* Final handover of:
  + Codebase
  + Architecture diagrams
  + Model and API documentation
  + Deployment & monitoring guide

**Final Deliverables**

* **Dashboard**: Real-time threat feeds, visual analytics, APM charts
* **AI Module**: Detects anomalies in threats (IP, domain behavior)
* **Backend + Ingestion**: Secure APIs with real/batch ingestion
* **APM**: Latency, throughput, uptime
* **Docs**: Swagger UI, deployment + user manuals
* **Cloud Deployment**: Dockerized, CI/CD-ready stack on AWS/GCP