Step No.	Task Name	Objective	Detailed Steps
1	Define the Scope and Objectives	Establish clear project goals and expected outcomes	- List threat types (e.g., phishing, malware) - Define outcomes (alerts, reports) - Outline key metrics and data points
2	Set Up the Development Environment	Prepare your coding environment for efficient workflow	- Install Python, Node.js, and Git - Set up virtual environments (venv for Python) - Install basic packages (Flask, scikit-learn, etc.) - Initialize Git
3	Project Structure Creation	Organize your project files systematically	- Create main directories: backend/, frontend/, data/, models/, scripts/ - Add essential files in each directory - Review structure for scalability
4	Initial GitHub Repository Setup	Version control setup and initial project documentation	- Create a GitHub repository - Push your local setup to GitHub - Add a basic README.md with project overview
5	Data Collection and Preprocessing	Gather necessary data and clean it for use in your project	- Collect datasets relevant to threats - Clean data (remove duplicates, handle missing values) - Feature engineering (create useful features)
6	Model Selection and Training	Choose suitable ML models for detecting cyber threats	- Research potential models (e.g., Random Forest, Neural Networks) - Train models on preprocessed data - Evaluate model performance
7	Backend Development (API Setup)	Develop the server-side application to handle requests and serve data	- Set up Flask server - Create endpoints for threat analysis - Integrate model predictions with backend logic
8	Frontend Development	Build the user interface for interaction and data visualization	- Set up React application - Design UI components for input, analysis results, and visualizations - Ensure responsive and user-friendly design
9	Integration and Testing	Connect frontend and backend, and ensure the system works as expected	- Integrate Flask APIs with React frontend - Perform unit and integration tests - Validate end-to-end functionality
10	Deployment to Vercel	Deploy the application for public access and testing	- Deploy frontend (React) on Vercel - Deploy backend (Flask) using Vercel serverless

Step No.	Task Name	Objective	Detailed Steps
			functions - Configure environment variables and API keys
11	Monitoring and Optimization	Monitor application performance and optimize for speed and accuracy	- Set up monitoring tools (e.g., Google Analytics, Sentry) - Optimize ML models for faster predictions - Improve UI/UX based on user feedback
12	Documentation and Finalization	Complete project documentation and prepare for handoff or further updates	- Write detailed documentation for users and developers - Prepare a final project report - Plan next steps for scaling or adding features