```
cyber_incident_monitoring/
│
├── frontend/              # Frontend for user interface
│   ├── public/            # Static assets (images, icons)
│   ├── src/
│   │   ├── components/     # React/Vue components
│   │   ├── pages/          # Individual pages (dashboard, incidents)
│   │   ├── services/       # API calls to backend
│   │   ├── App.js          # Main app component
│   │   ├── index.js        # Entry point
│   │   ├── styles.css      # Custom CSS
│   └── package.json        # Frontend dependencies
│
├── backend/               # Backend for data handling and API endpoints
│   ├── api/
│   │   ├── incidents/      # Routes for incidents handling
│   │   │   ├── __init__.py
│   │   │   ├── views.py     # API views for incidents
│   │   ├── users/          # Routes for user management (optional)
│   │   ├── ml/             # Machine learning API routes
│   ├── services/          # Services like scraping, data fetching
│   │   ├── data_pipeline.py   # Data pipeline management
│   │   ├── twitter_scraper.py # Scraping data from Twitter API
│   │   ├── news_scraper.py    # Scraping news data
│   ├── utils/             # Helper functions
│   │   ├── database.py      # Database connection handling
│   │   ├── preprocessing.py   # Data preprocessing scripts for ML
│   ├── app.py             # Main Flask or FastAPI application
│   ├── requirements.txt     # Backend dependencies
│
├── ml_model/              # Machine learning related files
│   ├── data/
│   │   ├── raw/            # Raw data collected from sources
│   │   ├── processed/       # Processed/cleaned data for ML model
│   ├── models/            # Saved ML models
│   │   ├── incident_classifier.pkl  # Trained ML model for incident classification
│   ├── notebooks/            # Jupyter notebooks for experimentation
│   ├── train_model.py         # Model training script
│   ├── evaluate_model.py      # Model evaluation script
│
├── database/                 # Database configuration and scripts
│   ├── migrations/           # Migration files (if using ORM like SQLAlchemy)
│   ├── init_db.py            # Script to initialize the database
│   ├── schema.sql            # SQL schema for relational database
│   ├── mongo_config.py        # MongoDB configuration
│
├── data_pipeline/            # Data collection and processing pipeline
│   ├── jobs/
│   │   ├── ingest_data.py     # Script to schedule data ingestion (scraping, API)
│   ├── logs/                 # Logs for pipeline activities
│   ├── airflow_dag.py         # Airflow DAG for data pipeline orchestration
│
├── config/                   # Configuration files for different environments
│   ├── config.py             # Main configuration (API keys, DB connections)
│   ├── development.yaml       # Development-specific settings
│   ├── production.yaml        # Production-specific settings
│
├── tests/                    # Unit and integration tests
│   ├── test_api.py           # Test cases for API routes
│   ├── test_ml.py            # Test cases for ML model predictions
│
├── deployment/               # Deployment configuration
│   ├── docker-compose.yml     # Docker Compose for local development
│   ├── Dockerfile            # Dockerfile for backend
│   ├── nginx/                # NGINX configuration for production
│   ├── k8s/                  # Kubernetes manifests (if using Kubernetes)
│
├── docs/                     # Documentation for the project
│   ├── README.md             # Overview of the project
│   ├── architecture_diagram.png   # Architecture diagram of the system
│   ├── flowcharts/           # Flowcharts and diagrams for understanding flow
│
```

```
├── .env                    # Environment variables for the project
├── .gitignore              # Files to ignore in version control
└── README.md                  # Main README for the project
```