Enchanted wings: marvels of butterfly species

# Introduction

* + **ProjectTitle:**[TrafficTelligence:AdvancedTrafficVolumeEstimationwith Machine Learning]
  + **TeamMembers:S.Revathi,SUmmeSalma,SNagendraprasad, v.subramani**
  + ​
  + **ProjectOverview**
  + **Purpose:Hereisacompletedraftofyourprojectdocumentationbasedonthe structure you provided:**
  + ​
  + ​
  + **---**
  + ​
  + 🚦**TrafficTelligence:AdvancedTrafficVolumeEstimationwithMachine Learning**
  + ​
  + **TeamMembers:**
  + **S.Revathi,S.UmmeSalma,S.NagendraPrasad,V.Subramani**
  + ​
  + ​
  + **---**
  + ​
  + 📘**ProjectOverview**
  + ​
  + **Purpose**
  + ​
  + **TrafficTelligenceisasmarttrafficmonitoringandestimationsystemdesigned to improve urban traffic management. Leveraging machine learning models, the system analyzes real-time traffic feeds and historical data to estimate vehicle volume, identify congestion patterns, and support predictive traffic planning.**
  + ​
  + **Features**
  + ​
  + **Real-time traffic volume estimation using ML models**
  + ​
  + **Interactive dashboard for visualizing traffic data**
  + ​
  + **Role-based authentication for users and admins**
  + ​
  + **API endpoints for feeding camera or sensor data**
  + ​
  + **Historical data analytics and visualization**
  + ​
  + **Alerts for abnormal traffic patterns**
  + ​
  + ​
  + ​
  + **---**
  + ​
  + 🏗**Architecture**
  + ​
  + **Frontend(React)**
  + ​
  + **Developed using React.js with functional components and hooks**
  + ​
  + **UIcomponentsdesignedwithMaterial-UI(MUI)foraresponsiveandmodern look**
  + ​
  + **React Router used for seamless page navigation**
  + ​
  + **Axios used for API calls to backend services**
  + ​
  + ​
  + **Backend (Node.js + Express.js)**
  + ​
  + **RESTful API built with Express.js**
  + ​
  + **Handlesauthentication,traffic dataprocessing,and MLmodel integration**
  + ​
  + **Implementsmiddlewareforerrorhandlingandtoken-basedaccesscontrol**
  + ​
  + ​
  + **Database (MongoDB)**
  + ​
  + **MongoDBused tostore trafficdata, userinfo, andsession logs**
  + ​
  + **Collections:**
  + ​
  + **users:Storesusercredentialsandroles**
  + ​
  + **traffic\_data: Logs of vehicle count, timestamps, location metadata**
  + ​
  + **alerts:Storescongestionorabnormalityreports**
  + ​
  + ​
  + ​
  + ​
  + **---**
  + ​
  + ⚙**SetupInstructions**
  + ​
  + **Prerequisites**
  + ​
  + **Node.js >= 16.x**
  + ​
  + **MongoDB installed locally or cloud-hosted (MongoDB Atlas)**
  + ​
  + **Python 3.9+ (for ML model integration via script or API)**
  + ​
  + ​
  + **Installation**
  + ​
  + **# 1. Clone the repository**
  + **gitclonehttps://github.com/your-username/traffictelligence.git**
  + **cd traffictelligence**
  + ​
  + **# 2. Install dependencies**
  + **cd client**
  + **npm install**
  + ​
  + **cd ../server**
  + **npm install**
  + ​
  + **#3.Setupenvironmentvariables**
  + **#Createa.env filein/server with:**
  + **MONGODB\_URI=<your\_mongo\_uri>**
  + **JWT\_SECRET=<your\_secret\_key>**
  + **PORT=5000**
  + ​
  + ​
  + **---**
  + ​
  + 📂**FolderStructure**
  + ​
  + **Client (React Frontend)**
  + ​
  + **client/**
  + **│**
  + **├── public/ #Static assets**
  + **├──src/**
  + **│├── components/ # Reusable UI components**
  + **│├── pages/ #ViewslikeDashboard,Login,etc.**
  + **│├── services/ # API handlers via Axios**
  + **│├── context/ # Authentication context**
  + **│└── App.js #Entry point**
  + ​
  + **Server (Node.js Backend)**
  + ​
  + **server/**
  + **│**
  + **├── controllers/ #Request handlers**
  + **├── models/ # Mongoose schemas**
  + **├── routes/ #APIroutedefinitions**
  + **├── middleware/ #Authanderrormiddleware**
  + **├── services/ # Business logic and ML model integration**
  + **├── .env #Environmentvariables**
  + **└── server.js # Entry point**
  + ​
  + ​
  + **---**
  + ​
  + 🚀**RunningtheApplication**
  + ​
  + **Start Backend**
  + ​
  + **cd server**
  + **npm start**
  + ​
  + **StartFrontend**
  + ​
  + **cd client**
  + **npm start**
  + ​
  + **EnsureMongoDBisrunningandenvironmentvariablesareconfigured properly.**
  + ​
  + ​
  + **---**
  + ​
  + 📡**APIDocumentation**
  + ​
  + **Base URL: <http://localhost:5000/api>**
  + ​
  + **Example: POST /auth/login**
  + ​
  + **Request:**
  + **{**
  + **"email": ["user@example.com",](mailto:user@example.com)**
  + **"password": "securepass"**
  + **}**
  + ​
  + **Response:**
  + **{**
  + **"token":"jwt\_token\_here",**
  + **"user": {**
  + **"id": "userId",**
  + **"role":"admin"**
  + **}**
  + **}**
  + ​
  + ​
  + **---**
  + ​
  + 🔐**Authentication**
  + ​
  + **Authentication is handled using JWT tokens**
  + ​
  + **Uponlogin, atoken isgenerated andstored inlocal storage**
  + ​
  + **Routesareprotectedwithmiddlewarethatverifiesthetoken**
  + ​
  + **Roles(e.g.,user,admin)controlaccesstocertainresources**
  + ​
  + ​
  + ​
  + **---**
  + ​
  + 🖥**UserInterface**
  + ​
  + **Screenshots/GIFs(Insertyourmedia here)**
  + ​
  + **DashboardView:Trafficvolumegraphsandlivecamerafeedintegration**
  + ​
  + **LoginPage:Secureloginwithform validation**
  + ​
  + **AlertPage:Visuallistofrecentalertsandnotifications**
  + ​
  + ​
  + ​
  + **---**
  + ​
  + 🧪**Testing**
  + ​
  + **TestingStrategy**
  + ​
  + **Unittestingwith Jestforbackend routesand controllers**
  + ​
  + **ComponenttestingwithReactTestingLibrary**
  + ​
  + **Manual integration testing for ML model predictions**
  + ​
  + ​
  + ​
  + **---**
  + ​
  + 📸**ScreenshotsorDemo**
  + ​
  + **> [Live Demo Link (if hosted)]**
  + **or**
  + **Attachscreenshotsof:**
  + ​
  + **Dashboard with traffic charts**
  + ​
  + **Login/Register pages**
  + ​
  + **Real-time data input and output**
  + ​
  + ​
  + ​
  + ​
  + ​
  + **---**
  + ​
  + 🐞**KnownIssues**
  + ​
  + **MLmodelperformance maydropunder lowlighting conditions**
  + ​
  + **Highlatencyobservedwhen processinglargevideo inputs**
  + ​
  + **Frontendformvalidation canbe bypassedwithout backendstrict checks**
  + ​
  + ​
  + ​
  + **---**
  + ​
  + 🌱**FutureEnhancements**
  + ​
  + **Integratereal-timevideofeedprocessingviaOpenCV**
  + ​
  + **DeploymodeltocloudusingTensorFlow.jsorFlaskmicroservice**
  + ​
  + **Add admin panel for user management**
  + ​
  + **Implementmobile-responsivedesign**
  + ​
  + **Improvepredictionaccuracywithmoretrainingdata**
  + ​
  + ​
  + ​
  + **---**
  + ​
  + **Letmeknowifyou'dlikethisinMarkdown,PDF,orasaGitHub README.md file!**
  + ​
  + ​
  + **Features:**Highlightkeyfeaturesandfunctionalities.

# Architecture

* + **Frontend:Developedusing React.jswithfunctional componentsand hooks**
  + ​
  + **UIcomponentsdesignedwithMaterial-UI(MUI)foraresponsiveandmodern look**
  + ​
  + **React Router used for seamless page navigation**
  + ​
  + **Axios used for API calls to backend services**
  + ​
  + **Backend:** RESTful API built with Express.js
  + ​
  + Handlesauthentication,traffic dataprocessing,and MLmodel integration
  + ​
  + Implements middleware for error handling and token-based access control
  + ​
  + **Database:**MongoDB usedto storetraffic data,user info,and session logs
  + ​
  + Collections:
  + ​
  + users: Stores user credentials and roles
  + ​
  + traffic\_data:Logsofvehicle count,timestamps,location metadata
  + ​
  + alerts: Stores congestion or abnormality reports

# Setup Instructions

* + **Prerequisites:**Node.js>=16.x
  + ​
  + MongoDB installed locally or cloud-hosted (MongoDB Atlas)
  + ​
  + Python 3.9+ (for ML model integration via script or API)
  + ​
  + **Installation:** # 1. Clone the repository
  + gitclonehttps://github.com/your-username/traffictelligence.git
  + cd traffictelligence
  + ​
  + # 2. Install dependencies
  + cd client
  + npm install
  + ​
  + cd ../server
  + npm install
  + ​
  + # 3. Set up environment variables
  + # Create a .env file in /server with:
  + MONGODB\_URI=<your\_mongo\_uri>
  + JWT\_SECRET=<your\_secret\_key>
  + PORT=5000
  + ​
  + ​
  + ---

# Folder Structure

* + **Client:** client/
  + │
  + ├── public/ # Static assets
  + ├── src/
  + │├── components/ # Reusable UI components
  + │├── pages/ #ViewslikeDashboard,Login,etc.
  + │├── services/ #API handlers via Axios
  + │├── context/ # Authentication context
  + │└── App.js #Entry point
  + ​
  + **Server:** server/
  + │
  + ├── controllers/ #Request handlers
  + ├── models/ # Mongoose schemas
  + ├── routes/ #API route definitions
  + ├── middleware/ # Auth and error middleware
  + ├── services/ # Business logic and ML model integration
  + ├── .env # Environment variables
  + └── server.js #Entry point

# Running the Application

* + Start Backend
  + ​
  + cd server
  + npm start
  + ​
  + Start Frontend
  + ​
  + cd client
  + npm start
  + ​
  + EnsureMongoDB is running and environment variables are configured properly.
  + ​
  + ​
  + ---

1. **Frontend:**cdclient
2. npmstart

* **Backend:**cdserver
* npmstart

# API Documentation

* + <http://localhost:5000/api>
  + ​
  + Example: POST /auth/login
  + ​
  + Request:
  + {
  + "email": ["user@example.com",](mailto:user@example.com)
  + "password": "securepass"
  + }
  + ​
  + Response:
  + {
  + "token": "jwt\_token\_here",
  + "user": {
  + "id": "userId",
  + "role": "admin"
  + }
  + }

# Authentication

* + Authentication is handled using JWT tokens
  + ​
  + Upon login, a token is generated and stored in local storage
  + ​
  + Routes are protected with middleware that verifies the token
  + ​
  + Roles(e.g.,user,admin)controlaccesstocertainresources
  + ​

# User Interface

Screenshots/GIFs(Insertyourmediahere)

DashboardView:Trafficvolumegraphsandlivecamerafeedintegration Login Page: Secure login with form validation

AlertPage:Visuallistofrecentalertsandnotifications

Ni

---

# Testing

* + TestingStrategy
  + ​
  + Unit testing with Jest for backend routes and controllers
  + ​
  + ComponenttestingwithReactTestingLibrary
  + ​
  + Manual integration testing for ML model predictions
  + ​

# ScreenshotsorDemo

* + [Live Demo Link (if hosted)]
  + or
  + Attach screenshots of:
  + ​
  + Dashboardwithtrafficcharts
  + ​
  + Login/Register pages
  + ​
  + Real-time data input and output
  + ​

# Known Issues

* + ML model performance may drop under low lighting conditions
  + ​
  + Highlatencyobservedwhen processinglargevideo inputs
  + ​
  + Frontend form validation can be bypassed without backend strict checks
  + ​

# FutureEnhancements

* + Integrate real-time video feed processing via OpenCV
  + ​
  + DeploymodeltocloudusingTensorFlow.jsorFlaskmicroservice
  + ​
  + Add admin panel for user management
  + ​
  + Implement mobile-responsive design
  + ​
  + Improve prediction accuracy with more training data
  + ​
  + ​
  + ​
  + ---