**Serenova: A Community-Driven Safety and Healing Platform**

* **Introduction**

Serenova is a web-based application designed to address two critical issues: preventing sexual violence and supporting survivors. The app combines real-time safety features (like safe route planning and harassment reporting) with

a supportive community space for survivors to share their stories, access resources, and heal. Serenova empowers users to navigate safely while also providing a safe space for survivors to connect, heal, and advocate for change.

* **Problem Statement**
  + Sexual violence is a pervasive issue, especially in countries like India, where women and vulnerable individuals often feel unsafe in public spaces.
  + Survivors of sexual violence often lack access to immediate support, legal aid, and counselling.
  + There is no centralized platform that combines prevention, reporting, and healing in one place.
* **Objectives**
* Prevention:
  + Provide safe route planning using real-time data.
  + Enable anonymous harassment reporting and create a heatmap of high-risk areas.
* Support:
  + Offer a safe space for survivors to share their stories and access resources.
  + Connect survivors with counsellors, legal aid, and support groups.
* Community Engagement:
  + Encourage community involvement through volunteer networks and educational content.
* **Features**
* **4.1 Safe Route Planner**
  + AI-Powered Safe Routes: Suggests the safest route based on real-time crime data, street lighting, and crowd density.
  + Safety Scores: Each route is assigned a safety score.
  + Live Location Sharing: Users can share their live location with trusted contacts.
  + Emergency SOS Button: Alerts emergency contacts and local authorities.
* **4.2 Harassment Reporting and Heatmap**
  + Anonymous Reporting: Users can report incidents of harassment, which are tagged on a real-time heatmap.
  + Community Alerts: Sends safety alerts when users enter high-risk areas.
  + Data for Advocacy: Aggregated data is shared with local authorities and NGOs.
* 4.3 Survivor Storytelling and Healing
  + Anonymous Storytelling: Survivors can share their stories anonymously.
  + Healing Resources: Access to counselling, legal aid, and self-care resources.
  + Support Groups: Moderated, anonymous support groups for survivors.
  + Advocacy Tools: Survivors can advocate for change by sharing their stories.
* 4.4 Community Engagement
  + Volunteer Network: Users can sign up to be "Serenova Volunteers" for safety patrols or emotional support.
  + Educational Content: Modules on consent, gender equality, and self-defence.
  + Crowdsourced Safety Tips: Users can share safety tips and strategies.
* **Technology Stack**
  + Frontend:
    1. React.js: For building a dynamic and responsive user interface.
    2. Material-UI/Tailwind CSS: For pre-designed, customizable UI components.
    3. Socket.io: For real-time features like live location sharing and SOS alerts.
  + Backend:
    1. Node.js: For server-side logic.
    2. Express.js: For creating RESTful APIs.
    3. JWT (JSON Web Tokens): For user authentication.
    4. Bcrypt: For encrypting sensitive user data.
* Folder Structure :
* ├── backend/ # Backend (Node.js/Express.js)
* │ ├── config/ # Configuration files (e.g., database, JWT)
* │ │ └── db.js # MongoDB connection setup
* │ ├── controllers/ # Logic for handling routes
* │ │ ├── authController.js # Authentication logic
* │ │ ├── routeController.js # Safe route planning logic
* │ │ ├── reportController.js # Harassment reporting logic
* │ │ └── storyController.js # Survivor story logic
* │ ├── middleware/ # Custom middleware (e.g., authentication)
* │ │ └── auth.js # JWT authentication middleware
* │ ├── models/ # MongoDB models (schemas)
* │ │ ├── User.js # User schema
* │ │ ├── Report.js # Harassment report schema
* │ │ └── Story.js # Survivor story schema
* │ ├── routes/ # API routes
* │ │ ├── authRoutes.js # Authentication routes
* │ │ ├── routeRoutes.js # Safe route routes
* │ │ ├── reportRoutes.js # Harassment report routes
* │ │ └── storyRoutes.js # Survivor story routes
* │ ├── utils/ # Utility functions
* │ │ └── helpers.js # Helper functions (e.g., encryption, validation)
* │ ├── app.js # Main backend application file
* │ └── server.js # Server setup and start
* │
* ├── frontend/ # Frontend (React.js)
* │ ├── public/ # Static assets (e.g., index.html, favicon)
* │ ├── src/ # React source code
* │ │ ├── assets/ # Images, icons, fonts
* │ │ ├── components/ # Reusable components
* │ │ │ ├── Navbar.js # Navigation bar
* │ │ │ ├── Footer.js # Footer
* │ │ │ ├── RoutePlanner.js # Safe route planner component
* │ │ │ ├── ReportForm.js # Harassment reporting form
* │ │ │ └── StoryCard.js # Survivor story card component
* │ │ ├── pages/ # Page components
* │ │ │ ├── Home.js # Home page
* │ │ │ ├── RoutePage.js # Safe route planning page
* │ │ │ ├── ReportPage.js # Harassment reporting page
* │ │ │ ├── StoriesPage.js # Survivor stories page
* │ │ │ └── Profile.js # User profile page
* │ │ ├── context/ # React context for global state
* │ │ │ └── AuthContext.js # Authentication context
* │ │ ├── hooks/ # Custom React hooks
* │ │ ├── services/ # API service calls
* │ │ │ ├── authService.js # Authentication API calls
* │ │ │ ├── routeService.js # Safe route API calls
* │ │ │ ├── reportService.js # Harassment report API calls
* │ │ │ └── storyService.js # Survivor story API calls
* │ │ ├── App.js # Main App component
* │ │ ├── index.js # Entry point for React app
* │ │ └── styles/ # Global styles (CSS/SCSS)
* │ └── package.json # Frontend dependencies
* │
* ├── ai-ml/ # AI/ML (Python)
* │ ├── models/ # Trained models
* │ ├── scripts/ # Python scripts
* │ │ ├── route\_optimization.py # Safe route optimization script
* │ │ └── sentiment\_analysis.py # Survivor story sentiment analysis script
* │ └── requirements.txt # Python dependencies
* │
* ├── database/ # Database-related files
* │ ├── migrations/ # Database migration scripts (optional)
* │ └── seeds/ # Seed data for testing (optional)
* │
* ├── tests/ # Test files
* │ ├── backend/ # Backend tests
* │ └── frontend/ # Frontend tests
* │
* ├── .env # Environment variables (e.g., API keys, database URL)
* ├── .gitignore # Files/folders to ignore in Git
* ├── package.json # Backend dependencies
* ├── README.md # Project documentation
* └── LICENSE # Project license (optional)
* **Database:**
  + MongoDB: For storing user data, incident reports, and survivor stories.
  + Mongoose: For defining schemas and interacting with MongoDB.
* **AI/ML:**
  + Python: For building AI/ML models.
  + TensorFlow/Scikit-learn: For route optimization and sentiment analysis.
  + Google Maps API/Mapbox: For route planning and heatmap visualization.
* **Real-Time Features:**
  + Socket.io: For real-time communication (e.g., live location sharing, SOS alerts).
* **Security:**
  + HTTPS: For secure communication.
  + End-to-End Encryption: For securing sensitive data.
  + Helmet.js: For securing Express.js apps.
* **Third-Party APIs:**
  + Google Maps API/Mapbox: For route planning and geolocation.
  + Twilio: For sending SMS alerts during emergencies.
* **Workflow**
* User Registration/Login:
  + Users sign up or log in using email or social media accounts.
* Safe Route Planning:
  + Users input their destination and receive the safest route.
* Harassment Reporting:
  + Users can anonymously report incidents, which are added to the heatmap.
* Survivor Storytelling:
  + Survivors can share their stories and access healing resources.
* Community Engagement:
  + Users can join volunteer networks, participate in safety patrols, or contribute to educational initiatives.
* **Factors Considered for Safe Route Planning**

**1. Real-Time Crime Data**

* **What It Is:** Data about recent crimes (e.g., theft, harassment, assault) in the area.
* **How It’s Used:** Routes with fewer reported crimes are considered safer.
* **Source:** Local police databases, crowdsourced reports from Serenova users.

**2. Street Lighting**

* **What It Is:** The presence and quality of streetlights along the route.
* **How It’s Used:** Well-lit streets are considered safer, especially at night.
* **Source:** Google Street View, OpenStreetMap, or crowdsourced data.

**3. Crowd Density**

* **What It Is:** The number of people present in the area.
* **How It’s Used:** Routes with moderate crowd density are preferred—too few people can be unsafe, while too many can be chaotic.
* **Source:** Real-time data from location-based services (e.g., Google Maps, Foursquare).

**4. Historical Incident Reports**

* **What It Is:** Data about past incidents of harassment or violence in the area.
* **How It’s Used:** Areas with fewer historical incidents are considered safer.
* **Source:** Serenova’s own database of user-reported incidents.

**5. Public Transport Availability**

* **What It Is:** The availability of public transport options (e.g., buses, metro) along the route.
* **How It’s Used:** Routes with accessible public transport are considered safer, especially at night.
* **Source:** Public transport APIs (e.g., Google Transit).

**6. Pedestrian-Friendly Infrastructure**

* **What It Is:** The presence of sidewalks, crosswalks, and pedestrian signals.
* **How It’s Used:** Routes with better pedestrian infrastructure are considered safer.
* **Source:** OpenStreetMap, Google Maps.

**7. Time of Day**

* **What It Is:** The time when the user is traveling (e.g., day vs. night).
* **How It’s Used:** Some routes may be safe during the day but unsafe at night.
* **Source:** User input or real-time clock data.

**8. User Preferences**

* **What It Is:** Custom preferences set by the user (e.g., avoiding dark alleys, preferring main roads).
* **How It’s Used:** Routes are tailored to the user’s specific safety concerns.
* **Source:** User profile settings in Serenova.
* **How Safety Scores Are Calculated**

1. **Data Collection:**
   * Gather data from multiple sources (e.g., crime databases, crowdsourced reports, public APIs).
2. **Weighted Scoring:**
   * Assign weights to each factor based on its importance (e.g., crime data may have a higher weight than street lighting).
   * Example Weights:
     + Crime Data: 40%
     + Street Lighting: 20%
     + Crowd Density: 15%
     + Historical Incidents: 10%
     + Public Transport: 10%
     + Pedestrian Infrastructure: 5%
3. **AI/ML Model:**
   * Use machine learning algorithms (e.g., decision trees, neural networks) to analyze the data and calculate a **safety score** for each route.
4. **Route Ranking:**
   * Rank routes based on their safety scores and recommend the safest one to the user.

* **Example Scenario**

**User Input:**

* **Start Point:** Home
* **Destination:** Office
* **Time of Day:** Night

**Data Analysis:**

1. **Route A:**
   * Crime Data: Low
   * Street Lighting: Good
   * Crowd Density: Moderate
   * Historical Incidents: Few
   * Public Transport: Available
   * Pedestrian Infrastructure: Excellent
   * **Safety Score:** 95/100
2. **Route B:**
   * Crime Data: Moderate
   * Street Lighting: Poor
   * Crowd Density: Low
   * Historical Incidents: Some
   * Public Transport: Unavailable
   * Pedestrian Infrastructure: Poor
   * **Safety Score:** 60/100

**Recommendation:**

* Serenova recommends **Route A** because it has a higher safety score.
* **Implementation Plan**
* Phase 1: Research and Planning
  + Conduct surveys and interviews to understand user needs.
  + Study existing apps to identify gaps.
* Phase 2: Design
  + Create wireframes and mock-ups using Figma or Adobe XD.
  + Design the user interface and user experience.
* Phase 3: Development
  + Set up the backend (Node.js, Express.js, MongoDB).
  + Develop the frontend (React.js, Material-UI/Tailwind CSS).
  + Integrate AI/ML models for route optimization and sentiment analysis.
* Phase 4: Testing
  + Test the app for bugs and usability issues.
  + Conduct user testing to gather feedback.
* Phase 5: Deployment
  + Deploy the app using Heroku, Netlify, or Vercel.
  + Monitor performance and user feedback post-launch.
* **Expected Outcomes**
  + Reduced Risk of Sexual Violence: Users can navigate safely using the safe route planner.
  + Empowered Survivors: Survivors have access to resources and a supportive community.
  + Awareness and Advocacy: Heatmap data drives policy changes and community action.
  + Community Engagement: Volunteers and educational initiatives create safer spaces.
* **Future Enhancements**
  + Mobile App: Develop a mobile version for iOS and Android.
  + AI Chatbot: Provide real-time emotional support and guidance.
  + Gamification: Introduce rewards for safe behaviour and community contributions.
  + Integration with Wearables: Sync with smartwatches for real-time safety alerts.
* **Conclusion**

Serenova is a comprehensive solution to the problem of sexual violence, combining prevention, support, and community engagement in one platform. By leveraging cutting-edge technology and fostering a supportive

community, Serenova aims to create a safer and more inclusive world for everyone.