|  |
| --- |
| **Project Title:** Personalized Safety Routing System for Pedestrians and Cyclists Using Multi-Factor Risk Assessment and Machine Learning |
| **Project Summary:**  **What We're Doing**  We are developing an intelligent, machine learning-powered navigation system that prioritizes user safety alongside travel efficiency for pedestrians and cyclists in New York City. Unlike traditional navigation apps that focus solely on shortest routes, our system integrates multiple risk factors including crime statistics, traffic accidents, infrastructure quality, user-reported hazards, and contextual conditions to generate personalized safety scores for route segments and recommend the safest possible paths.  **Why This Matters**  Current navigation systems like Google Maps treat all streets as equally safe, potentially directing vulnerable users through high-crime areas, dangerous intersections, or poorly maintained infrastructure without their knowledge. This gap is particularly critical for women traveling alone at night, elderly pedestrians, and cyclists navigating areas with inadequate bike infrastructure. With NYC's Vision Zero initiative emphasizing safer streets and expanding cycling infrastructure, there's an urgent need for navigation tools that actively consider safety factors in routing decisions.  **Target Users**  Our primary users include urban pedestrians and cyclists who prioritize personal safety, particularly:   * Commuters navigating unfamiliar neighborhoods * Women traveling alone, especially during evening hours * Elderly pedestrians seeking well-lit, accessible routes * Recreational cyclists exploring new areas * Delivery workers and rideshare drivers serving vulnerable populations * Urban planners and transportation authority’s seeking data-driven safety insights   **Example Use Case**  Sarah, a graduate student at NYU, needs to walk home from the library at 10 PM through Brooklyn. Instead of taking the shortest route through areas with recent crime incidents, our system analyzes her personal safety preferences, current conditions (time of day, weather), and real-time data to recommend a well-lit path with good pedestrian infrastructure. The system shows her three route options: the shortest (12 minutes, moderate safety), the safest (16 minutes, high safety), and a balanced option (14 minutes, good safety). Sarah selects the balanced route, arriving safely while adding only two minutes to her commute. The system learns from her choice to better personalize future recommendations.  This vision transforms urban navigation from purely efficiency-focused to safety-conscious, empowering users to make informed decisions about their routes while contributing to a comprehensive understanding of urban safety patterns. |
| **Project Development:**  **Project Development: Minimal Viable Product (MVP)**  **MVP Description**  Our MVP will be a web-based safety routing application focused on Manhattan with basic crime-aware routing capabilities. Users can input origin-destination pairs and receive three route options: shortest path, safest path, and balanced path. The system will integrate NYPD crime data and basic infrastructure information to generate simple safety scores for route segments.  **Core MVP Features:**   * Basic web interface for route input and display * Integration of NYPD crime data (2022-2024) for Manhattan * Simple safety scoring algorithm using crime density * Route comparison showing safety scores vs. travel time * User feedback collection for route preferences   **Learning Objectives & Measurements**  **Primary Learning Goals:**   1. **User Acceptance:** Do users prefer safety-aware routes over shortest paths? 2. **Safety Perception:** How accurately do our safety scores match user perceptions? 3. **Efficiency Trade-offs:** What safety-efficiency balance do users find acceptable?   **Key Metrics:**   * Route selection patterns (% choosing safe vs. fast routes) * User satisfaction ratings (1-7 Likert scale) * Safety score accuracy (correlation with user-reported safety perceptions) * Time penalty acceptance (average additional minutes users accept for safety) * System usability scores (SUS methodology) * Geographic usage patterns across Manhattan neighborhoods   **System Architecture**   * The system will have a simple web setup. * The part users see will be separate from the part that handles data and calculations.   What Users Will See   * The website will be built with React.js and work on computers and phones. * The main feature will be a map (using Leaflet.js or Mapbox) showing routes and safety information. * A section will show different route options so users can pick the one they like. * Sliders will let users choose whether safety or speed is more important. * Simple forms will let users give feedback or report dangerous spots.   Server Side   * Flask or Django will be used to make an API for route calculations. * Map and location data will be stored in PostgreSQL with PostGIS. * Python libraries like NetworkX and OSMnx will be used to calculate routes. * Simple machine learning with scikit-learn will give routes a safety score based on crime and accident data. * Redis will store some data temporarily to make the website faster. * Users can create accounts to save their preferences.   Data Sources   * Crime data from the NYPD API to know where incidents happen. * Street and road information from OpenStreetMap using OSMnx. * Car accident records from NYC open datasets. * Users can report hazards directly through the app. * Weather data will be used since rain or snow can make routes riskier.   **Development Methodology: SCRUM**  **Sprint Structure (2-week sprints):**   * **Sprint 1-2: Data pipeline and basic routing**   + Set up development environment and database   + Get NYPD data and process it   + Build a simple shortest-path routing algorithm * **Sprint 3-4: Safety scoring and web interface**   + Create a crime-based safety scoring model   + Build React frontend with map integration   + Add route comparison feature * **Sprint 5-6: System testing and optimization**   + Deploy MVP (minimum viable product) to staging   + Test algorithm using historical NYC safety data   + Add monitoring tools to check performance   **SCRUM Roles:**   * **Product Owner:** Defines what users need and sets acceptance criteria * **Scrum Master:** Helps the team, removes problems, and keeps sprints on track * **Development Team:** Full-stack developers and data scientists   **Key Meetings:**   * Daily standups (15 minutes) * Sprint planning (4 hours at the start of sprint) * Sprint reviews and retrospectives (2 hours each) * Backlog refinement (ongoing)   **Success Criteria for MVP:**   * 95% of Manhattan route requests work successfully * Average response time below 3 seconds * At least 70% user satisfaction * Clear patterns in user preferences * System ready to expand to Brooklyn and add more safety factors |
| **System Evaluation Plan:**  **Evaluation Method:**   * Use both numbers (quantitative) and feedback (qualitative) to measure system performance   **Quantitative Evaluation:**   * **Historical Data Check:** Compare our routes with Google Maps using old crime and accident data * **Algorithm Testing:** Measure route calculation speed, reliability, and scalability * **Safety Score Check:** Compare safety scores with expert opinions using past incident data * **Route Quality:** Look at route variety, safety vs speed balance, and city coverage   **Qualitative Evaluation:**   * **Expert Review:** Ask urban planners and safety professionals to check routes * **Literature Comparison:** Compare results with research and best practices * **Case Studies:** Look closely at high-risk and low-risk routes in NYC   **Using Evaluation for Development:**   * **Short-term (Sprints 4-6):** Use results to improve algorithms and adjust safety scoring * **Medium-term (Next quarter):** Decide which features to add or improve based on evaluation * **Long-term:** Use results to plan expansion, work with transport authorities, and scale system to other cities |
| **Project Management Plan**  **How We Manage the Project:**   * Use SCRUM methodology with GitHub Projects * **GitHub setup:**   + Kanban boards with Backlog, In Progress, Review, Done   + Milestones for major deliverables   + Issues with user stories, priority, and effort estimates   + Pull requests linked to issues for code review   + Track progress with burndown charts and velocity insights   **Team Structure:**   * 2-week sprints with clear goals * Daily standups via Slack with GitHub updates * Sprint planning using GitHub milestones * Code review required before merging code * Continuous integration with GitHub Actions   **Key Deadlines:**   * Phase 1 (Weeks 1-6): MVP Development   + Week 2: Data pipeline and database ready   + Week 4: Basic routing and safety scoring done   + Week 6: Web interface deployed and tested * Phase 2 (Weeks 7-12): Testing and Validation   + Week 8: Safety scoring validated with historical data   + Week 10: Compare routes against baseline   + Week 12: Optimize performance and run statistical checks * Phase 3 (Weeks 13-16): Analysis and Documentation   + Week 14: Complete system evaluation   + Week 16: Submit final deliverables and documentation   **Success Metrics**  **Technical Success:**   * 95% uptime during testing * Routes calculated in less than 3 seconds on average * No critical security issues * All core algorithms fully tested   **User Success:**   * Safety algorithm accurately scores routes using historical data * Routes are safer compared to baseline routing * Users get different route options that balance safety and speed * Algorithm works well across different NYC areas   **Project Success:**   * Deliverables completed on time * Project completed under budget * Methodology is clear and reproducible * Results and documentation are ready for publication |

|  |
| --- |
| **Team Name: Group 2** |
| **Team Members:**   |  |  |  | | --- | --- | --- | | **Name** | **Student Number** | **Contact Number** | | Huda Ibraheem | D24126339 | 083 447 7463 | | Karan Joseph | D24125555 | 087 492 4547 | | Shalini Kuruguntla | D24126048 | 089 213 4144 | | Sai Priyanka Basa Shanker | D24125575 | 089 976 1927 | | Hina Kausar | D24127853 | 0899484178 | |
| **Team Meetings:**  Our team consists of five members, and we decided from the start that regular communication was important to the project's success. Initially, we formed a WhatsApp group to provide quick updates and arrange the availability. After our initial in-person meeting at the University, we decided to formalise our approach by scheduling daily Microsoft Teams call until mid of December. These calls are intended to deliver progress reports, to address problems, and to plan future stories and tasks requires for the JIRA board. We have formalized project board on JIRA and recording our code base and documentations in GitHub public repository.  All members are actively encouraged to attend meetings, however we recognise that conflicts may emerge from time to time. In such circumstances, participants provide updates to the WhatsApp group to ensure that we are moving forward smoothly. Meetings are usually held online for convenience, however face-to-face sessions will be held as needed (for example, milestone planning or presentations).  Decisions are made collaboratively, whenever possible, unanimous consent is preferred, but if consensus cannot be reached, majority voting is employed. Expertise in specific areas is also valued, for example - those with more technical understanding may help guide coding & framework selections. Also, turn-taking in conversations is handled informally, with each person given a time to speak before conclusions are formed. We are also recording our work together as show and tell demo which we planned to store on YouTube channel as well. |
| **Team Conflict:**  So far, our group has been performing well and without serious problems. However, we are aware that conflicts or individual availability routines may cause problems later. To avoid issues, we decided to maintain open and courteous interactions at all times. If someone is unable to attend a meeting or is delayed in finishing a work, the member is obliged to advise the team via messaging or chat so that expectations are clear.  In the event of remaining issues, we shall first try to address them directly during group sessions by encouraging open conversation. If tensions worsen, we will enlist the module coordinator in mediation. Our approach to conflict resolution will be based on compromise and ensuring that all views are heard. To avoid disagreements, we intend to allocate jobs evenly based on strengths and availability. No single member has veto power; choices will be made by a majority vote, promoting fairness and communal responsibility of outcomes. |