

CSE470: Software Engineering  
Section: 5  
Group: 3  
**SPRINT 1**

|  |  |
| --- | --- |
| Sadat Mahmud | 22301301 *(Scrum Master)* |
| Rana Mustafa | 21101060 |
| Atoshi Samadder | 21301706 |
| Nabil Nashit | 21201060 |

**Software: VisiNexus –** Connecting insights through vision

**Requirements**

1. Real-time demos for computer vision applications such as object detection, emotion recognition, etc.
2. Interactive tools for learning and understanding computer vision applications
3. Documentation and tutorials
4. API access and integration guide
5. User Management
6. Virtual coins for purchasing accessibility

**Framework Setup**

|  |  |
| --- | --- |
| Frontend | React.js |
| Backend | Express.js |
| Computer Vision Models | OpenCV and TensorFlow |
| Database | MongoDB |
| Hosting | GoDaddy |

**Workload Distribution**

|  |  |
| --- | --- |
| Sadat Mahmud | Project Manager: Oversee environmental setup, and proper deployment |
| Rana Mustafa | Computer Vision Specialist: Optimize Computer Vision Models |
| Atoshi Samadder | Frontend Developer: Develop the user interface |
| Nabil Nashit | Backend Developer: Build the server-side logic and API endpoints |

**Projected Features**

1. **User Management and Authentication**

**Objective:** Ensure secure user access with role-based functionalities and maintain user data integrity.

**Key Features:**

1. **User Registration and Login:**
   * Secure Registration: Collect user information (username, email, password) with validation to ensure correct formats.
   * JWT-based Authentication: Use JSON Web Tokens to manage user sessions, store tokens securely (e.g., HttpOnly cookies), and refresh tokens to maintain secure access.
   * Password Encryption: Hash passwords using bcrypt to prevent password leaks if data is compromised.
2. **Role-Based Access Control:**
   * User Roles: Define at least two roles: “User” and “Admin.”
     + User: Limited access to the core functionalities (e.g., demos and basic analytics).
     + Admin: Access to advanced analytics, settings, and model management sections.
   * Permissions: Route and component-level protection to ensure only authorized users access certain features.
3. **Profile Management:**
   * Profile Information: Allow users to update profile details, including usernames, profile pictures, and preferences.
   * API Usage Preferences: Enable users to configure notifications or limits for API usage, such as reminders or alerts for usage thresholds.
   * Password Management: Implement “Forgot Password” and “Change Password” functionalities using secure token-based reset links.
4. **Data Analytics and Visualization**

**Objective:** Provide insightful analytics to improve understanding of user engagement, model performance, and resource usage.

**Key Features:**

1. **User Dashboard:**
   * Basic Account Information: Show personal details, profile picture, and preferences.
   * Recent Activity: Display a log of recent demos used, with details like date, model, and processing time.
   * API Usage Tracking: Display the number of API calls made, with visual indicators for usage limits (e.g., bars or counters).
2. **Analytics Dashboard:**
   * Overall, API Usage: Track the total number of API calls, categorized by demo type (object detection, classification, emotion recognition).
   * Average Processing Time: Measure and display the average processing time for each model and demo type to highlight performance.
   * Data Visualizations:
     + Usage Trends: Display historical trends of API usage, such as usage spikes over time.
     + Performance Metrics: Visualize model accuracy, prediction distribution, and confidence intervals, giving insights into model reliability and accuracy.
   * Export Options: Allow admins to export analytics data in CSV or JSON format for further analysis or reporting.
3. **Model Performance Metrics:**
   * Accuracy Metrics: Show metrics for each model type, such as accuracy, precision, recall, and F1 scores.
   * Processing Time: Measure and display the average and max processing time for each model.
   * Demo Logs: Log each demo session, capturing details like model type, confidence score, and timestamp, to allow for detailed post-demo analysis.

**Rate Limiting:**

1. **API Usage Restrictions:**
   * Per-User Rate Limits: Set limits on the number of requests per minute, hour, or day per user to avoid overloading.
   * Notification System: Notify users when they approach or exceed limits, with options to request higher limits for specific use cases.
   * Admin Control: Allow admins to adjust rate limits for different users or models based on demand and resource availability.
2. **Computer Vision Functionality**

**Objective**: Deliver a robust and interactive computer vision experience with various models to showcase AI capabilities.

**Key Features:**

1. **Object Detection:**
   * Real-time Detection: Use the webcam feed to capture live images or allow users to upload images.
   * Bounding Box Display: Visualize detected objects with bounding boxes, labeling each detected object with a name and confidence score.
   * Model Selection: Provide options for different object detection models (e.g., YOLOv3 for fast detections, SSD for smaller object detection).
   * Error Handling: Manage errors gracefully, like unsupported file formats or model timeouts, to provide a smooth user experience.
2. **Image Classification:**
   * Classification Categories: Users can classify uploaded images into predefined categories such as animals, objects, or scenes.
   * Confidence Score Display: Show a detailed breakdown of classification results with confidence levels for each category.
   * Classification History: Maintain a history of recent classifications for users to review past results or experiment with different images.
3. **Emotion Recognition:**
   * Emotion Categories: Recognize and categorize emotions from facial images (e.g., happy, sad, angry, surprised).
   * Interactive UI: Provide an interactive UI for users to upload images or use the webcam, displaying results with corresponding confidence scores.
   * Potential Use Cases: Include examples of potential applications, like customer engagement or mood tracking, to emphasize the technology's practical uses.

**Model Selection and Customization:**

1. **Model Options:**
   * Selection Interface: Implement a dropdown or selection panel for users to choose between models tailored to different tasks (e.g., object detection vs. emotion recognition).
2. **Customizable Model Parameters:**
   * Parameter Controls: Include sliders or input fields for adjusting parameters like detection threshold, model size (fast vs. accurate), or batch processing.
   * Real-time Feedback: Show immediate changes in model output as parameters are adjusted to allow users to experiment and understand how these settings impact results.
3. **Additional Documentation and Support**

**Objective:** Provide users and developers with comprehensive guides to understand and use the platform effectively.

**Key Features:**

1. **API Documentation:**
   * Endpoint Overview: Provide a list of API endpoints, with descriptions for each feature (e.g., object detection, classification).
   * Example Requests and Responses: Show sample requests and responses for each endpoint, with explanations for each field.
   * Error Codes and Troubleshooting: List common error codes and troubleshooting tips to help users resolve issues.
2. **User Guide:**
   * Feature Walkthrough: Provide a step-by-step guide on using each demo feature.
   * FAQ Section: Include common questions and answers about using the platform, customizing settings, and understanding model outputs.
3. **Developer Guide:**
   * Setup Instructions: Explain how to set up the project for local development, including dependencies and environment configuration.
   * Custom Model Integration: Guide developers on how to integrate custom models or modify existing ones.
   * Contribution Guidelines: Detail guidelines for contributing to the codebase, including coding standards, review process, and documentation requirements.