**Clickbait prevention extension. (real-time)- cybersecurity integrated with AI**

**Objective:- Malvertising – malicious advertising**

**Time Estimates for Learning and Development**

1. **Python Programming (NumPy and Pandas)**:
   * **Time**: 1 week
   * **Purpose**: Enhance your data manipulation skills for preprocessing data.
2. **Web Scraping (BeautifulSoup)**:
   * **Time**: 1 week
   * **Purpose**: Collect a dataset of headlines for training your model.
3. **Natural Language Processing (NLP)**:
   * **Time**: 2 weeks
   * **Purpose**: Preprocess and analyze text data for feature extraction.
   * **Key Topics**: Tokenization, stopwords removal, TF-IDF.
4. **Machine Learning (Logistic Regression)**:
   * **Time**: 2 weeks
   * **Purpose**: Build and train a simple classification model.
   * **Key Topics**: Training and evaluating models, accuracy, precision, recall, F1-score.
5. **API Development (Flask)**:
   * **Time**: 1 week
   * **Purpose**: Create a RESTful API to handle predictions.
   * **Key Topics**: Creating and handling API requests.
6. **JavaScript and Browser Extensions**:
   * **Time**: 2 weeks
   * **Purpose**: Develop a basic browser extension that integrates with your API.
   * **Key Topics**: Basic JavaScript, Chrome Extension APIs.
7. **User Interface (HTML/CSS)**:
   * **Time**: 1 week
   * **Purpose**: Design a simple user-friendly interface for the browser extension.
   * **Key Topics**: Basic HTML/CSS for styling the extension's popup.
8. **Version Control (Git and GitHub)**:
   * **Time**: 1 week
   * **Purpose**: Manage and collaborate on your project.
   * **Key Topics**: Git basics, using GitHub.

**Development and Integration**

* **Time**: 2 weeks
* **Purpose**: Integrate all components, test, and debug the system.
* **Activities**: Assembling the project, testing the model and API, refining the browser extension, user interface tweaks.

**Total Estimated Time:**

**12 weeks** (3 months)

**Essential Topics for Demonstration Project**

**1. Python Programming**

* **Purpose: Used for data processing, model training, and API development.**
* **Key Topics: Basic syntax, functions, error handling, and libraries (NumPy, Pandas).**

**2. Web Scraping**

* **BeautifulSoup:**
  + **Purpose: To collect a small dataset of headlines.**
  + **Key Topics: Parsing HTML, extracting data.**

**3. Natural Language Processing (NLP)**

* **Purpose: To preprocess and analyze text data (headlines) for model training.**
* **Key Topics: Tokenization, stopwords removal, TF-IDF for feature extraction.**

 **TF (Term Frequency)**: Measures how frequently a term appears in a document.

 **IDF (Inverse Document Frequency)**: Measures how important a term is by considering how often it appears across all documents.

 **TF-IDF**: Combines TF and IDF to give a balanced representation of a term's relevance in a document.

**4. Machine Learning**

* **Purpose: To build and train a simple classification model.**
* **Key Topics: Logistic Regression, model evaluation (accuracy, precision, recall, F1-score).**

**5. API Development**

* **Flask:**
  + **Purpose: To create a RESTful API that allows real-time predictions.**
  + **Key Topics: Creating and handling API requests.**

**6. JavaScript and Browser Extensions**

* **Purpose: To develop a basic browser extension that integrates with your API.**
* **Key Topics: Basic JavaScript, Chrome Extension APIs.**

**7. User Interface (HTML/CSS)**

* **Purpose: To create a simple user-friendly interface for the browser extension.**
* **Key Topics: Basic HTML/CSS for styling the extension's popup.**

**8. Version Control**

* **Git and GitHub:**
  + **Purpose: To manage and collaborate on your project.**
  + **Key Topics: Git basics, using GitHub.**

**Workflow and Timeline**

**Week 1-2: Initial Setup and Web Scraping**

* **Developer 1: Set up Python environment, enhance NumPy and Pandas skills, start web scraping.**
* **Developer 2: Set up project structure, learn basic JavaScript, design initial UI for extension.**

**Week 3-4: Data Collection and API Development**

* **Developer 1: Complete web scraping, preprocess data, save to CSV.**
* **Developer 2: Develop Flask API, define endpoints, begin integration with extension.**

**Week 5-6: Data Preprocessing and NLP**

* **Developer 1: Tokenize, remove stopwords, extract features using TF-IDF.**
* **Developer 2: Continue developing and refining the browser extension, integrate with API.**

**Week 7-8: Model Training and UI Design**

* **Developer 1: Train the Logistic Regression model, evaluate performance, save model and vectorizer.**
* **Developer 2: Finalize UI design for the browser extension, ensure responsiveness and usability.**

**Week 9-10: Integration and Testing**

* **Developer 1: Document data processing, NLP, and model training steps.**
* **Developer 2: Test and debug API endpoints, integrate and test browser extension.**

**Week 11-12: Final Adjustments and Documentation**

* **Both Developers: Collaborate on final testing, debugging, and ensuring seamless integration.**
* **Developer 1: Finalize documentation for data and model processes.**
* **Developer 2: Finalize documentation for API and front-end processes.**

**Work division for 2**

**Developer 1: Data and Model Specialist**

**1. Python Programming (NumPy and Pandas)**

* **Enhance data manipulation skills for preprocessing data.**
* **Time: 1 week**

**2. Web Scraping (BeautifulSoup)**

* **Task:**
  + **Learn BeautifulSoup and write a script to scrape headlines from multiple sources.**
  + **Preprocess the collected data and save it to a file (e.g., CSV).**
* **Deliverables:**
  + **A cleaned dataset of headlines stored in a CSV file.**
* **Time: 1 week**

**3. Natural Language Processing (NLP)**

* **Task:**
  + **Perform tokenization, stopwords removal, and TF-IDF feature extraction.**
  + **Prepare the data for model training.**
* **Deliverables:**
  + **Preprocessed and vectorized headline data.**
* **Time: 2 weeks**

**4. Machine Learning (Logistic Regression)**

* **Task:**
  + **Train and evaluate the clickbait detection model.**
  + **Save the trained model and vectorizer using pickle.**
* **Deliverables:**
  + **Trained model and vectorizer saved as files.**
  + **Model evaluation metrics documented.**
* **Time: 2 weeks**

**5. Version Control (Git and GitHub)**

* **Task:**
  + **Manage and collaborate on the project's codebase.**
  + **Commit and push code changes to the repository.**
* **Deliverables:**
  + **Regularly updated and maintained repository.**
* **Ongoing**

**6. Documentation**

* **Task:**
  + **Create documentation for data processing, NLP, and model training steps.**
  + **Document the model evaluation and performance results.**
* **Deliverables:**
  + **Comprehensive documentation for the data and model processes.**
* **Time: 1 week**

**Developer 2: API and Front-End Specialist**

**1. API Development (Flask)**

* **Task:**
  + **Create a RESTful API to handle predictions.**
  + **Define API endpoints and handle requests.**
* **Deliverables:**
  + **Functional Flask API with endpoints for predictions.**
* **Time: 1 week**

**2. JavaScript and Browser Extensions**

* **Task:**
  + **Develop a basic browser extension.**
  + **Integrate the extension with the Flask API to fetch and display predictions.**
* **Deliverables:**
  + **Working browser extension integrated with the Flask API.**
* **Time: 2 weeks**

**3. User Interface (HTML/CSS)**

* **Task:**
  + **Design a simple user-friendly interface for the browser extension's popup.**
  + **Ensure the extension's UI is responsive and intuitive.**
* **Deliverables:**
  + **User-friendly and responsive UI for the browser extension.**
* **Time: 1 week**

**4. Version Control (Git and GitHub)**

* **Task:**
  + **Manage and collaborate on the project's codebase.**
  + **Commit and push code changes to the repository.**
* **Deliverables:**
  + **Regularly updated and maintained repository.**
* **Ongoing**

**5. Testing and Debugging**

* **Task:**
  + **Test the API endpoints and ensure they return correct predictions.**
  + **Test the browser extension functionality and fix any issues.**
* **Deliverables:**
  + **Fully tested and debugged API and browser extension.**
* **Time: 1 week**

**6. Documentation**

* **Task:**
  + **Create documentation for API development and integration steps.**
  + **Document the browser extension's functionality and user interface design.**
* **Deliverables:**
  + **Comprehensive documentation for the API and front-end processes.**
* **Time: 1 week**