**Project Proposal**

**Project Scope**

The proposed project, **Named Entity Recognition (NER) System for User Data Extraction**, aims to automatically tag and extract relevant information from user-provided documents such as **resumes, CVs, or forms**. This tool will help users by identifying and categorizing essential details such as **names, phone numbers, email addresses, ages, education qualifications, and work experiences.** The system will leverage **Python** for backend processing and **NLP** capabilities, while the frontend interface will be built using **React** to ensure a responsive and user-friendly experience.

**Technologies and Methodology**

The project will utilize Python and React as its core technologies. Python will manage backend processing, particularly focusing on **Named Entity Recognition (NER)** functionality using **NLP libraries like spaCy or nltk**. These libraries will enable the identification and categorization of entities within input documents. A **Flask-based RESTful API** will facilitate communication between the frontend and backend, handling user-submitted documents and returning tagged information. On the frontend, **React** will be used to build a user-friendly interface, allowing users to upload documents and view the extracted and tagged information seamlessly.

**User Stories and Development Timeline**

The tool will be developed with a user-centric approach, driven by the following key user stories:

* As a **user**, I want to upload my resume or document and have key information automatically tagged (e.g., name, phone number, email, etc.) so that I don’t need to manually enter this data.
* As an **HR professional**, I want to extract essential candidate details quickly from multiple resumes to streamline the hiring process.
* As a **researcher**, I want to analyze documents for named entities like organizations, locations, and dates for research purposes.

**Development Timeline Using Agile Methodology**

The project will follow an Agile framework with 10 weeks divided into five **2-week sprints**. Each sprint will focus on developing specific features while incorporating feedback from the previous sprint.

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Week(s)** | **Objectives** |
| Sprint 1 | 1-2 | Project Setup & Research:  • Set up the Python and React development environment.  • Conduct research on NER techniques.  • Collect and preprocess the dataset.  • Initialize GitHub repository with README. |
| Sprint 2 | 3-4 | Basic NER Model Implementation:  • Integrate spaCy or nltk for basic entity recognition.  • Develop initial entity extraction rules.  • Start API development using Flask. |
| Sprint 3 | 5-6 | Frontend Development & Integration:  • Build the document upload feature using React.  • Connect frontend with backend Flask API.  • Implement display for extracted entities (tags for name, phone, email, etc.). |
| Sprint 4 | 7-8 | Advanced NER Customization:  • Improve entity extraction accuracy with custom models.  • Add support for industry-specific tags (e.g., job titles, skills).  • Conduct integration tests. |
| Sprint 5 | 9-10 | Final Testing & Deployment:  • Fix bugs and optimize performance.  • Perform final testing on multiple document formats.  • Prepare documentation and final project report.  • Final project presentation and demo. |