**1. Functional Requirements**

1. **Interactive Chatbot**
   * **Skill Input**: The system shall allow students to enter one or more skills (e.g., Java, Python, Project Management) via a chatbot interface.
   * **Course Recommendations**: Upon receiving the skill inputs, the chatbot shall retrieve and display a list of relevant courses that teach those skills.
   * **Job Recommendations**: The chatbot shall also provide the top 3–5 job roles (e.g., Software Developer, Data Analyst) that align with the user’s entered skill set.
   * **Data Integration**: The chatbot’s recommendations shall be based on data/analysis from the project’s notebook, web scraped csv’s, etc.
2. **Job Field Selection and Skills Graph**
   * **Dropdown Menu**: The website shall provide a dropdown menu listing available job fields (e.g., Enterprise Systems, Data Science, Software Development).
   * **Dynamic Graph**: On selection of a job field, the website shall display a graph (bar chart or similar) showing the top skills associated with that job field. Sunburst Chart created from data shows top skills associated with each job field.
   * **Course Matching**: Below the graph, the system shall list recommended courses that map to those top skills.
3. **Data Management**
   * **Skill/Job/Mapping Data**: The system shall maintain or access a structured dataset (or database) that maps skills to courses and skills to job roles from the backend of the website.
   * **Updates**: It shall be possible to update or refresh this data (e.g., if new courses are introduced or job role requirements change).
4. **User Interaction & Navigation**
   * **Clear Navigation**: Users shall be able to navigate between the chatbot feature and the job field dropdown easily.
   * **Filtering/Searching** (Optional Enhancement): The system could allow additional filtering or searching of courses/jobs by keywords if needed.
5. **Error Handling**
   * **Invalid Inputs**: If the chatbot receives unrecognized skills or incomplete data, the system shall display an informative message or suggest the closest match.
   * **Data Retrieval Errors**: In case of data unavailability, the website shall display an error message and prompt the user to try again later or contact support.

**2. User Interface Requirements**

1. **Homepage & Layout**
   * **Homepage**: Provide an overview of the site’s purpose, including a brief description of the chatbot and the dropdown menu feature.
   * **Consistent Design**: All pages shall follow a consistent design (color scheme, typography, layout) to ensure a cohesive user experience.
2. **Chatbot Interface**
   * **Text Input Box**: Clearly visible text input box for students to type in their skills or questions.
   * **Response Display**: Chatbot responses shall be displayed in a conversational style, showing recommended courses and top job roles.
   * **Quick-Reply Buttons** (Optional): Provide pre-set skill tags or quick suggestions for common skills to guide users.
3. **Job Field Dropdown & Chart**
   * **Dropdown Placement**: The dropdown menu for selecting a job field shall be clearly labeled and easily accessible (e.g., near the top or on a dedicated page/section).
   * **Chart Visualization**: A responsive chart (bar chart, stacked bar, or column chart) to show top skills required for the chosen job field.
   * **Course List**: Below or next to the chart, list the recommended courses that align with the displayed skills.
4. **Responsiveness & Accessibility**
   * **Responsive Design**: The website layout (including the chatbot and the chart) shall adapt to various screen sizes (desktop, tablet, mobile).
   * **Accessibility**: The website shall follow basic accessibility guidelines (e.g., proper color contrast, keyboard navigation, alt text for images, etc.).
5. **User Feedback & Help**
   * **Help/FAQ Section**: Provide a section or link for frequently asked questions and guidance on how to use the chatbot and dropdown features.
   * **Contact/Support** (Optional): Include a way for users to provide feedback or request additional assistance (e.g., an email address or contact form).

**3. Non-Functional Requirements**

1. **Performance**
   * **Response Time**: The chatbot should respond to user queries within a reasonable time (e.g., under 2 seconds for most queries).
   * **Scalability**: The system should handle concurrent user requests without significant degradation in performance (e.g., up to a specified number of users).
2. **Maintainability**
   * **Modular Architecture**: The codebase (frontend, backend, data handling) should be organized so that updates to course data or skill mappings do not require major code rewrites.
   * **Documentation**: Provide clear documentation for future developers or maintainers (including setup instructions, data structure explanations, etc.).
3. **Reliability & Availability**
   * **Uptime**: The website should aim for high availability (e.g., 99% uptime during semester hours).
   * **Error Logging**: The system shall log errors or exceptions in a manner accessible to the development/maintenance team for troubleshooting.
4. **Compatibility**
   * **Browser Support**: The website should function correctly on the latest versions of major browsers (Chrome, Firefox, Safari, Edge).
   * **Mobile Compatibility**: Key features (chatbot, dropdown menu, charts) should be usable on common mobile devices.
5. **Testing**
   * **Functional Testing**: Each feature (chatbot skill input, dropdown selection, chart display) should be tested to ensure correct behavior and data retrieval.
   * **User Acceptance Testing**: Conduct tests with sample student users to verify the site is intuitive and meets their needs.
   * **Performance & Load Testing**: Test system responsiveness under peak loads if usage is expected to be high.

**Additional Future Considerations**

* **Analytics & Tracking**: Track user interactions (e.g., popular skill searches, most recommended courses) to improve the system over time.
* **Internationalization**: If needed, provide language support for non-English speakers.
* **Integration**: Potential integration with student portals or learning management systems (e.g., auto-enrollment or direct links to course registration).