**Team** 4 **– NMTA (New Mexico Tech Advisor)**

**Project Proposal & Plan**

# **Introduction (20 points)**

We want to create a website that facilitates the process of scheduling courses for students at New Mexico Tech. The current official solution that Banweb offers can feel confusing to navigate and lacks visual demonstrations for student schedules. Currently, Banweb does not give students an indication of what their day by day schedule would look like before they register for all their courses. Based on anecdotal evidence that those in our team have experienced, students are often forced to use 3rd party sources such as Coursicle in order to visualize their semesters.

As similar projects to our idea have been developed before, we decided it would be best to expand upon the concept. Namely, we plan not only for our website to serve as a schedule visualizer, but also as an advisor of sorts. The advisor in this case would be a chatbot that students can interact with to receive information and advice on various teachers and courses that they may be interested in. We believe this will differentiate our project from those that have been done before, while giving us all an opportunity to learn about the training and deployment process of large language models, which are becoming increasingly more important in the field of computer science.

## **Project Overview and Statement of Proposal**

As briefly mentioned in the introduction, we plan on developing a website that allows users to create a visual representation of their college semester schedule in an aim to clarify the process of course registration. The main website features will consist of a calendar graphic, a search bar, and an LLM chat bot trained on New Mexico Tech course data. A user should be able to add a course using the search bar functionality, which will then be reflected on the calendar. The calendar will support overlapping for the class graphics, which will provide the user with an indication of any course conflicts. The chat bot will serve as an advisor. It will respond to any questions that the user may have about specific classes or instructors. It will be trained on scraped course data from the NMT catalog along with instructor data from a review site such as Rate My Professors. An example interaction would be the user asking about the software engineering course. They should be able to ask in plan text such as “tell me about software engineering”, and the chatbot should provide every instance of software engineering that is available during that semester along with the instructors attached. If more than one instructor exists, the chatbot should be able to suggest the most favorable one.

**Statement of Proposal:** *We propose to create a website that allows students at New Mexico Tech to create their class schedule with optional advice from an artificial intelligence advisor.*

## **Project Scope and Objectives**

The scope of the data set used for the website database and chatbot training will be limited to courses and instructors within the domain of the computer science department. We want to limit the capabilities of the chatbot to the aforementioned functionality for the time being as the difficulty scaling to make it more functional may cause a disproportionate amount of time, which could add additional risk elements to the project. For the purposes of demonstration, we want to host the website on a local machine. There it can be accessed via localhost and reduce the chances of something going wrong before or during the demo.

# **Risk Management Strategy (50 points)**

## **Risk Table**

| **Risks** | **Category** | **Probability** | **Impact** | **RMMM** |
| --- | --- | --- | --- | --- |
| Failing to Scrape Course Catalog Website | PR & ST | Unlikely | 3 | 2.3 |
| Failing to Scrape Rate My Professor | PR & ST | Unlikely | 2 | 2.3 |
| Failing to implement LLM Distribution | PR & DE | Unlikely | 2 | 2.3 |
| Lack of Python Web Server experience | ST | Very Unlikely | 4 | 2.3 |

**Category values: Impact values:**

PS – Product Size Risk 4 – catastrophic

BU – Business Impact Risk 3 – critical

CU – Customer Relations Risk 2 – marginal

PR – Process Risk 1 – negligible

TE – Technology Risk

DE – Development Environment Risk

ST – Risk Associated with Staff Size and Experience

## **Discussion of Risks to Be Managed**

All of the risks in the table involve failure to implement certain functionalities that we are not entirely sure are possible. Web scraping is something that our team is quite inexperienced in, hence the risks involved with scraping falling under the ST category. Other technical risks, such as website frontend and backend development, are not included as risks because our team is experienced in website creation. The reason all of our risks are unlikely to happen is that they all seem possible to implement given enough time. Since we have a relatively large amount of time to complete the project, it is unlikely that between the four of us, we will fail to find solutions.

## **Risk Mitigation, Monitoring, and Management Plan**

Perseverance and rigorous research will be necessary to avoid the risks in the table (Section 2.3.1). Time will be the main factor when monitoring risks (Section 2.3.2). If the risks do become problems, we have ideas of how the software should look with these issues in mind (Section 2.3.3).

### Risk Mitigation

The first two risks noted in the table involve web scraping large amounts of information to be used in our software. The risk is that we will simply be unable to implement web scrapers in time to be used in data collection. The best way we can avoid this is by starting early and using familiar tools that will help us scrape, i.e. Python. Similarly, we must have perseverance when performing research and implementation of the LLM distribution if we want our idea of the software to be fully realized. In regard to the last risk, it can be mitigated by the use of educational resources for developing using Python and Flask.

### Risk Monitoring

For all of the risks, time will be the major factor when monitoring if the risk is becoming an issue. Jaxen and Chris will pay particular attention to implementation time as they are the ones tasked with implementing the scrapers. If around two weeks pass without having implemented scraping, the risk will become a problem. For Rate-My-Professor scraping and LLM distribution, the entire team will keep an eye out on how much time is left to complete the project. Whether or not the implementations are doable given that timeframe is a decision that will be made by the entire group. In regard to the last risk, each member will be responsible for learning Flask themselves but the team as a whole will ensure that everyone is educated and work together to fill in gaps of knowledge.

### Risk Management (Contingency Plans)

If we are unable to implement course catalog scraping, we will need to manually input the data that we need. For the course catalog, this includes all classes, Professors, prerequisites, offering semesters, and so on. This will cause us to scale down our scope of how many majors the software will be implemented for. If course catalog scraping was not possible, it is likely that Rate-My-Professor is also not possible to scrape. In this case, the functionality of including Rate-My-Professor reviews will be omitted from the software. If the LLM distribution is not possible to implement, the functionality will also need to be omitted from the software. General information about the course can still be given if the previous two risks are not possible. We can include the course description, professor, and offering semester. As a last resort, if using Python and Flask is not plausible, then we will resort to using Java and Apache Tomcat as the source of our web server.

# **Schedule (20 points)**

Below is a list of tasks associated with the project, a Gantt chart depicting task durations, dependencies and completion dates, and a summary of resource assignments for each task. We wanted to use an Agile Scrum methodology for managing the project. As such, we created a Sprint outline in addition to the Gantt Chart in order to better represent our project schedule. Note that this means we have two different timelines, although we plan to follow the Agile framework.

## **Task List (Task Backlog) and Sprint Outline (**For Agile Methodology)

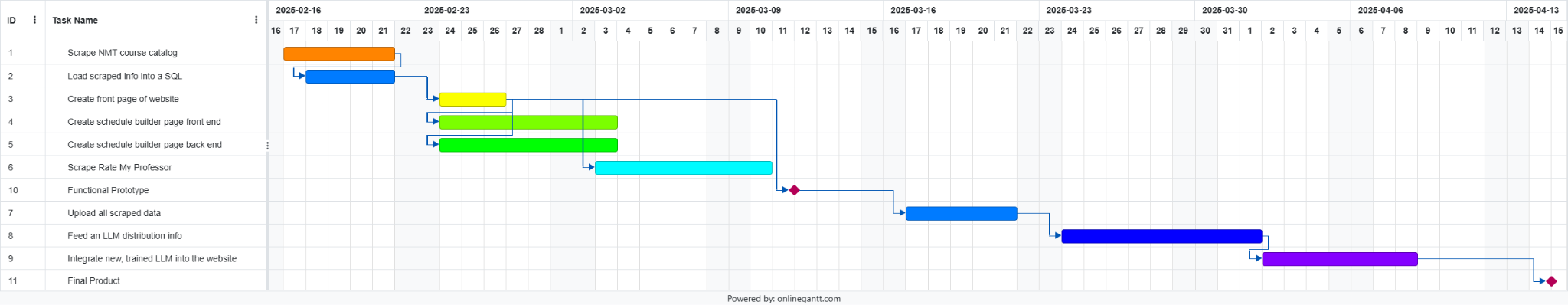
1. Scrape NMT course catalog website for Computer Science courses
2. Load scraped info into a SQL database
3. Create front page of website
4. Create schedule builder page front end
5. Create schedule builder page back end
6. Scrape Rate My Professor for information on each professor
7. Upload all scraped data into a text file
8. Feed an LLM distribution info from the text file to build a personality revolved around providing useful information for each class and professor
9. Integrate new, trained LLM into the website

Below is an outline of our project overview using an Agile Scrum framework. We decided to split the timeline into five Sprints, each of which will have an interval of two weeks. A point system will be used to indicate the effort required to complete each milestone. Note this is just an outline and a better visual representation will be provided in the future.

| Task Name | Story Points | Priority | Sprint | Assignee(s) | Status |
| --- | --- | --- | --- | --- | --- |
| Scrape NMT course catalog | 8 | Medium | Sprint 1 (Feb 16 - Mar 1) | Jaxen, Chris | In Progress |
| Load scraped info into database | 2 | Medium | Sprint 1 (Feb 16 - Mar 1) | Jaxen, Chris | To Do |
| Create front page | 3 | High | Sprint 1 (Feb 16 - Mar 1) | Tyler, Zixi | In Progress |
| Create schedule builder front end | 5 | High | Sprint 1 (Feb 16 - Mar 1) | Chris, Jaxen | To Do |
| Create schedule builder back end | 8 | High | Sprint 2 (Mar 2 - Mar 15) | Zixi, Tyler | To Do |
| Project Demo | 3 | High | Sprint 2 (Mar 2 - Mar 15) | Jaxen, Chris, Tyler, Zixi | To Do |
| Scrape RMP | 8 | Medium | Sprint 3 (Mar 16 - Mar 29) | Jaxen, Chris | To Do |
| Upload scraped data into file | 2 | Medium | Sprint 3 (Mar 16 - Mar 29) | Jaxen, Chris | To Do |
| Feed LLM data | 8 | High | Sprint 4 (Mar 30 - Apr 12) | Zixi, Tyler | To Do |
| Integrate LLM | 13 | High | Sprint 5 (Apr 13 - Apr 26) | Jaxen, Chris, Tyler, Zixi | To Do |
| Product Finalization and demo | 3 | High | Sprint 5 (Apr 13 - End of semester) | Jaxen, Chris, Tyler, Zixi | To Do |

Sprint 5 will extend past the usual two week interval of each sprint. We decided to do this instead of creating a 6th Sprint as any tasks assigned at this point would be previously unfinished tasks. A new Sprint would be redundant in this case.

## **Timeline Chart (**Gantt Chart for Proposal Requirement)



1. (Feb 16 - Feb 22): Scrape NMT course catalog
2. (Feb 18 - Feb 22): Load scraped info into a SQL
3. (Feb 23 - Feb 26)Create front page of website
4. (Feb 27 - March 5) Create schedule builder page front end
5. (Feb 27 - Mar 5) Create schedule builder page back end
6. (Mar 3 - Mar 10)Scrape Rate My Professor
7. (Mar 9) **Functional Prototype**
8. (Mar 12 - Mar 15) Upload all scraped data
9. (Mar 17 - Mar 21) Feed an LLM distribution info
10. (Mar 24 - Apr 1) Integrate new, trained LLM into the website
11. (Apr 2 - Apr 13)Integrate new, trained LLM into the website
12. (Apr 14)**Final Product**

## **Resource Table**

| **Task** | **People** | **Hardware & Software** | **Special** |
| --- | --- | --- | --- |
| Task #1 | Jaxen, Chris | PC, Python, VSCode |  |
| Task #2 | Jaxen, Chris | PC, Python, VSCode, SQL |  |
| Task #3 | Jaxen, Chris, Tyler, Zixi | PC, Python, HTML, CSS, Flask, VSCode |  |
| Task #4 | Jaxen, Chris | PC, Python, HTML, CSS, Flask VSCode |  |
| Task #5 | Tyler, Zixi | PC, Python, HTML, CSS, Flask, VSCode |  |
| Task #6 | Jaxen, Chris | PC, Python, VSCode |  |
| Task #7 | Jaxen, Chris | PC, Python, VSCode |  |
| Task #8 | Zixi, Tyler | PC, Python, VSCode | LLM Distribution, unknown model at this time |
| Task #9 | Jaxen, Chris, Tyler, Zixi | PC, Python, HTML, CSS, Flask, VSCode | LLM Distribution, unknown model at this time |

# **Project Resources (10 points)**

## **People**

1. Tyler Del Prete
2. Jaxen Bujold
3. Zixi Li - Team Leader
4. Christopher Esquibel

## **Hardware and Software Resources**

* Personal Computer, laptop, desktop, etc.
* Coding environment (VSCode, Eclipse)
* Python, HTML, CSS, Flask languages
* MySQL and MySQL Workbench
* Microsoft Excel/Google Sheets (for Sprint management)
* GitHub

## **Special Resource**s

* LLM Distribution (undecided, but likely Llama 3.1 8B or Llama 3.2 3B)

# **Appendices (5 points)**

Team meetings will occur on the Monday, Wednesday, and Friday of each week at 1:00 PM. Each meeting should last anywhere between 30 minutes to 1 hour. Any tasks not finished during the current Sprint will be moved to the next Sprint at the first available meeting. The meetings will conclude on May 6, 2025 or at an unspecified earlier date assuming all requirements for the completion of the project are met beforehand.

**5.1 Proposal Contributions**

Section 1 was done by Zixi

Section 2 was done by Jaxen

Section 3 was done by Tyler

Section 4 was done by Christopher