SENIOR PROJECT PROPOSAL

CSE 499

ERICK VEGA

1. Project Name

   EnrollEase (Smart Class Registration)

2. Group Size / Contact information

   Erick Vega

3. Abstract/Propose

   - Minimize the time required for students to register for classes.

   - Minimize walk-in distance from class to class.

   - Empower students to generate customized schedules.

   This is a web application dedicated to students that will enable them to choose a set of classes, select their schedule preferences, select an optimal schedule and register from a single button.

4. Background/Prior Knowledge

   - I know Python, HTML, CSS, and a little bit of JavaScript which are the main tools that I will be using for my application. Since I don't have too much experience with full-stack applications, I can consider myself a Newbie.

   - My web app will be developed in Django, a web development framework written in Python in addition to Bootstrap for the front-end. There are certainly many web applications developed in Django, however, I haven't seen a product similar to my idea. I gained experience working on microservices from a previous internship that will help me develop the back-end part of my application, however, I haven't worked on user interfaces or databases.

   - My idea was born when I found myself manually arranging all my classes at BYU-I in the most optimal way to have all my classes in the morning. Later, my idea gained strength when I took a Discrete Math class which increased my curiosity about using graphs for optimal path finding; I think that with that knowledge I can optimize a schedule to minimize the walking distance of students from class to class, for example.

   - When presenting my idea in an Architectural Design class, everyone agreed that my idea was not just great, but innovative, and marketable. Having this type of support from other students confirms that they are the right audience and encourages me to trust that my application will solve their class registration challenges being a meaningful project to work on.

5. Description

   - **Why?** – During the last three years I have manually "optimized" my classes to avoid walking too much or making sure they don't collide with each other. It is a real pain having to register every semester, and most students would agree. Registering for classes and choosing a schedule doesn't have to take more than a couple of minutes, right now it can take hours, if you want a "smart" schedule. I have the student experience and the technology to automate the registration process, why don't make it real and useful for everyone?

   - **What?**: My web-app is simple and intuitive, it will generate an optimized schedule for a student's classes and will let him register from a single button. The app will have integration with RateMyProfessor which will facilitate choosing classes based on professor's reviews from other students. A solution for this problem doesn't exist yet, so creating such a solution will reduce the time students take to register for classes, reduce the time they walk on campus, and increase their productivity. If the app can generate an optimal schedule in seconds, that is all that students need to never go back to the old "manual" process of registering for classes.

   - **Who?**: Any student in the world who has to choose his/her classes frequently is the perfect customer for this application. We cannot guarantee always finding the best option for the student.

   - **Where?** This is going to be a web-application compatible with major web browsers and mobile devices. The application will be deployed on AWS or Heroku.

   - **How?** A student logs in with his school account, selects the classes he/she wants for a specific semester, the student selects schedule preferences (i.e. Only classes on MWF, only remote classes, only classes in the morning), the app generate a set of optimal schedules, the student selects the best schedule for him/her, and register from one button to all the classes.

   - **When?** The product is ready when a student can select a set of classe and get at least one optimal schedule. I know that I can move on from one stage to another when the basic functionality is working; a nice user interface and advanced features can wait.

6. Significance

   - My project will solve a recurrent problem that nobody else has tried to solve which can potentially change the way registration for classes works. Thousand of students can benefit from this application, not only at BYU-Idaho, but at other registration-based institutions. Being the first product to solve this issue is significant because is a pioneer in an unknown field.

   - Yes, most companies look for full-stack developers. This project covers all the basic concept of full-stack development, APIs, databases, user interfaces, controllers, security, cloud services and more. In a resume, I can specify how many people I was able to help with my solution and how much time they saved using my application.

7. New Computer Science Concepts

The following are the concepts that I will need to learn to complete this project:

- Django: This web framework helps build a full-stack web application in Python.
- Bootstrap: This is a library that to help design websites or web-apps.
- SQLite databases: A type of database that works with Django.
- Authentication: Ability to have multiple users on an application based on their credentials to access the web-app
- Deploying on Cloud Services (AWS, Heroku):

8. Interestingness
   - This project is interesting for me because I am relying on my math, software, and creative skills to complete this project, I haven't had the opportunity to combine all mentioned skills to develop a product. I am interested in seeing my friends using it to register for classes, especially my brother who will be a first-year student soon. I am interested on building my first web-app and understand how it really works in the back-end.

9. Tasks and Schedule
   - **Week 1: Research & Development (12 hours)**
     - Research and gain insights into Django and Bootstrap.
   - **Week 2-3: Requirements Analysis (20 hours)**
     - User Interface Design: 7 hours
     - Database Setup: 5 hours
     - User Authentication: 5 hours
     - Class Selection: 3 hours
   - **Week 4-5: Implementation (20 hours)**
     - Bootstrap Integration: 4 hours
     - Django Basics: 6 hours
     - Schedule Preferences: 5 hours
     - Optimized Schedule Generation: 5 hours
   - **Week 6-7: Implementation (20 hours)**
     - RateMyProfessor Integration: 6 hours
     - Testing and Debugging: 14 hours
   - **Week 8-9: Test and Integration (20 hours)**

  - Testing and Debugging: 14 hours
  - Deployment: 6 hours
- **Week 10-11: Finalization and Documentation (20 hours)**
  - Documentation: 15 hours
  - Final Review and Presentation: 5 hours
- **Week 12-13: Buffer/Adjustments (8 hours)**
  - Reserve some time for any unexpected challenges or additional learning needs.
- Each week, I will create two lists for a specific task: one containing essential requirements (must-haves) and another with additional features (should-haves). I will consider the task complete once I have completed all the essential requirements from my must-have list.

10. Resources
    - Programming languages: Python, JavaScript, HTML, CSS, SQL
    - Frameworks: Django, Bootstrap
    - Database: SQLite3
    - Cloud services for deployment: AWS or Heroku (costs around $12 USD/year)
    - IDE: Visual Studio Code
    - Browser: Google Chrome
    - OS: Windows 11
    - Hardware: Extra Monitor ($300)