

MIS 5050/6050

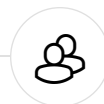
Web Application Project Description



Overview

The purpose of the Web Application Project is to allow you to *apply* what you have learned in the readings, assignments, and class work to a project of your own choosing. You will identify a business need, real or fictitious, for which you will design a functional Node.js application with a web interface and a database back-end. The project is a team effort in which you are expected to translate your knowledge of web technologies and methods into an applied product to solve a business problem or create a business opportunity. The project carries significant weight in the course grade and, if done well, will be an impressive addition to your professional portfolio.

The project is divided into milestones that will be due periodically throughout the semester. You will submit your milestone deliverables on Canvas. For two of the milestones, you will also meet briefly with the instructor outside of class to review your project progress and discuss problems as well as next steps. Due dates and specific requirements for each milestone are posted on Canvas.



Project Teams

You will complete the semester project in self-selected teams of 2 students (teams of 3 students may be formed with instructor approval in some cases, but teams of more than 3 students will not be approved and should be split into separate teams). You may handle division of responsibilities as you see fit; however, it is very important that each team member makes an equal contribution to the project. Failure of a team member to contribute will result in loss of points for that team member. You are encouraged to divide responsibilities in such a way that each team member has the opportunity to write code for a part of the application.



Project Scope

You will identify a business need/opportunity that could benefit from the creation of a data-driven web application. The project need not be for a "real" client in that it is not expected that you interview users, etc., in the course of the project; however, the type of organization and problem must be familiar to both the class and the instructor. It is possible for students with access to a real stakeholder, such as a family business or a job, to use a real client. Projects may include not-for-profit and government agencies as well. However, to ensure that your course grade is not jeopardized by stakeholder requirements, care should be taken to decouple the product delivered for the class from any product subsequently delivered to the business stakeholder. *Note that projects from other classes may only be used if the team is identical in both classes.*

The project is not envisioned as a sweeping implementation to manage the data needs of the entire organization. Instead, a fairly narrow set of *functions* should be identified for implementation. These functions could require a fairly broad data model, but the user interface will remain narrowly focused to the chosen tasks. It is expected, however, that the implementation be in depth and detailed for the narrow range of functions implemented. Implementation standards are covered in greater detail below.

In addition to identifying business functions to be implemented in the project, the project will require the design and implementation of a database (either SQL or NoSQL) that provides the backing store for the application's interface. The database itself will not be graded per se; however, it should follow good database design principles and enable proper functioning of the application to fulfill its intended purpose.



Project Requirements

The application developed for your project must include a minimum level of Node.js components/features that we will cover in class. Specific minimum requirements are as follows:

1. 5 interlinked *dynamic* pages (views) that display dynamic, database-driven content **per team member**. **To be counted as dynamic, the page must retrieve information to and/or write information to the application database using one or more**

queries/data access functions written by the team.

2. At least 5 database transactions (a transaction includes selecting, inserting, updating, or deleting data from the database) **per team member**
3. Support for user authentication and secure storage of user credentials
4. Preservation of state (may include cookies, session state, url parameters, querystring, etc.)
5. Development of at least one REST API and consumption of another third-party API
6. Inclusion and use of at least two npm libraries that are *not* covered elsewhere in the course (i.e., you need to explore additional functionality on your own).
7. Input validation and error handling for all free-form user entries that could potentially result in errors

The requirements are necessarily broad in order to accommodate a wide range of applications. If you have questions about a requirement as it relates to your specific project, **see me early**. I am willing to be flexible on some of the requirements provided that you can demonstrate to me that your alternative constitutes a comparable level of functionality. Again, these are minimum requirements – you are encouraged to exceed them if your application warrants it. (But, please don't go overboard).



Project Evaluation

Projects will be evaluated using the following rubric:

Objective/Criteria	Exceptional	Developing	Beginning
Website design and functionality (e.g. do all aspects of the site work as intended and does the website accomplish its intended purpose)	All aspects of final product work as expected. Website supplies full support for all required operations and fulfills its intended purpose. (30 points)	Most aspects of final product work, but there are some errors or missing components. Website partially supports required functions. (20 points)	Final project is mostly inadequate for its intended purpose. Several missing operations or non-functional components. (10 points)

Required Node.js components	All required components present and functional. (30 points)	Most components present, but some missing or not functional (20 points)	Several missing or non-functional components (10 points)
Ease of Web Site Navigation	Website is easy to navigate with highly intuitive navigation controls. Pages are well organized with adequate user guidance. (15 points)	Navigation is possible, but is unintuitive or partially non-functional. Pages lack some organization or user guidance. (10 points)	Navigation is difficult. Navigation controls absent or non-functional. Little evidence of page organization or user guidance (5 points)
Look and feel including consistency	The final product is aesthetically pleasing, with a highly professional appearance and style. (15 points)	Evidence of some formatting, but lacking the polish of a professionally formatted project. (10 points)	Little evidence of formatting. Final deliverable lacked professional appearance. (5 points)
Documentation (including code commenting)	Documentation is complete and professionally done, including adequate code commenting. (10 points)	Documentation is partially complete, but lacking professionalism or missing some components (e.g. code commenting) (7 points)	Documentation is poorly done or missing several required components. (4 points)



Project Tips and Pointers

1. Try to choose a teammate with skills complementary to your own.
2. Don't procrastinate. Adopt the mantra of "steady incremental progress." This is not a trivial project and will be extremely difficult to "cram" in at the end of the semester.
3. Don't overextend yourself and attempt to tackle something overly complex for your first project. Keep your data model simple and straightforward (avoid numerous cross-table relationships). I will let you know if I think the scope of your project needs to be adjusted. It is easier to add additional functionality later than have to scale back part-way through.
4. Follow a structured and organized programming approach. Make sure your code is clear and easily understood with descriptive code commenting. Your aim should be to make your code clearly understandable to a novice developer.
5. I am available for help during office hours and by appointment throughout the semester. I am less sympathetic to students experiencing last-minute problems when I haven't seen or heard from them during the course of the semester.