Proposal

Part 1:

Our project Matchy is a platform for university students to find teammates for group projects. Instead of making a post on every course discussion board after the term begins, the users will be able to set their status on a unified profile of themselves and start to search for, and get searched by, potential teammates, even before the term begins. The platform will also encourage positive collaboration by displaying, as a part of the user profile, how others rate their collaboration experience with the student. In short, we are hoping to integrate finding suitable teammates and potential friends, team building and cooperation into a very smooth process on a single platform.

Part 2.

An explanation of how your website meets each of the feature specification requirements listed above (user profiles, user authentication, etc.)

a) User Profiling:
   User profiling is one important key in our project. Every user can see the classmates' profiles, which include many useful information like name, contact, grade, and so on. Among which, the most important one is evaluations from their past partners. This way, users have more available information when it comes to choose partners for important projects.

b) User Authentication and Authorization:
   This part is easy to be fulfilled, to be honest. Each user of course has his or her own credentials to log in. Our idea couldn't be implemented if the web app did not fulfill this requirement.
   Basically everyone, who uses this web app, needs to log in first to do whatever he or she needs to do. Of course, the admin will be logged into a different page, where the administrator can do his thing.

c) Data:
   We must have posts data, where user publishes to find partners.
   We also must have evaluations data, which is the key differentiation point of our web app.
   Of course, profile data, like name and contact. Also login data.

d) Navbar, footer, these basic elements of every modern website.
   But also, views like *login view*, *sign up view, profile view, project status view, skills view, evaluation view, message view,* and so on.

e) Admin:
   For this kind of platform, there has to be administrators so that when harmful or irrelevant content is posted, the admin can delete it. In addition, the admins have the role to update information like username, which users don't have access to change. There may be more features coming out as we go along.

Part 3.

What are all the different things a user will do when they use your web app?

1. Change status:  The user can change the status to searching for teammates for a certain assignment of a course, or indicate they don't want to team up with anyone so that they will not be included in the search result, or be invited,
2. Search for teammates:  The user can search all students who are looking for teammates for a certain assignment.
3. Invite teammates: When the user wants to work with another student, he/she can send an invitation to that student with a message. The student who receives the message can accept or decline the invitation.
4. Working with others: When the user finds all the teammates, he/she can start a project on the website which includes all the teammates to indicate they are working together.
5. Reviewing teammates: When a team finishes the project, all members can rate their teammates, from 1 to 5 stars. They can also leave a review message, which can be anonymous.

When will you need to pull in data from an outside source?

Basically, all user interactions require data from the database since all user information is stored in the database. For example, if a user wants to search for teammates, we would need to search the database to find all the qualified students.