

Assignment 5

Web Application on Sharing Economy

Due Dates: December 6, 11.59PM

In this assignment, you will design and implement a web application for a sharing economy of your choice. There is no specific project topic for you, and you must select the topic and the scope of your web application. So, be creative!

You should develop this web application in a team of four. Start early!

Overview

In this project, (i) first, you specify what you want to implement (e.g., a car rental platform, a room sharing platform, etc.), (ii) then you design the details of your application (e.g., the software architecture, web-page mock-ups, your test plan, project task allocation among team members) and (iii) finally you implement it. The phases are for your own management, you don't need to submit the phases, submit a complete project report at the end.

Although there is no specific project topic here, you have to deliver list of common features no matter what topic you choose. We provide two sample projects in the appendix. These are provided to inspire you. That is, you will not receive the creativity mark, if you choose any one of these two projects. Also, these projects are sample only, use your imagination to create an application that users can easily use.

Specification

- Your application should be built using NodeJS or any framework depended on NodeJS as backend programming language and you should try to host on any hosting server of your choice (e.g., Heroku provides free hosting). Follow this [tutorial](#) to get yourself familiar with Heroku.
- Your project must include the following features:
 - **Profiling:** You must design and implement a set of features representing user profiles. Using the profiling features, users can update their personal information, and can see the profile of other users.
 - **User Authentication and Authorization:** Your system must authenticate users based on passwords. Your application also needs to support at least one 3rd party authentication such as Google/Facebook/Github sign-in, or two-factor authentication. Users have different roles in your system, and you must authorize them when they are conducting certain tasks. For instance, a normal user cannot

change the password of other users, or update someone else's personal information.

- **Social Network:** In any sharing economy platform, we will have a social network. In AirBNB, hosts and guests form a social network; In Uber, drivers and passengers form a social network. In such networks, users can interact with one another: can review each other, provide references, and other users make decision based on those reviews and interactions.
- **Rating and Commenting:** When users use a shared resource provided by another user, they must be able to rate and comment on that offering. Using this commenting and rating features, others can decide whether to use that resource or not. For example, on AirBNB, a guest can review the host and the place. Others will read reviews and decide whether to rent that place or not.
- **Search and Recommendation System:** Your system should provide the basic functionalities to search the shared economy. Using this feature, users should be able to search and discover what they want to use. It is really important that you use a smart recommendation system that ranks the matching entries according to user's query and personal data. Moreover, using the same recommendation system, the system should show similar things when a user is browsing a certain item. For instance, when one is looking at a room on AirBNB, the system lists similar properties on the same page.
- **Admin:** You should provide basic admin functionalities, including but not limited to changing passwords, updating and deleting information, and initializing/repopulating the database.
- Note that you will get up to extra 10 points for any significant additional features.
- All feature functionality should have proper test cases with good comments in code.
- You should take care of at least two well known security vulnerability, using a framework is fine. In your project report, you should document how this requirement is meet
- Once you have finished the first version of your project, measure the performance. Apply performance improvement techniques and show how much performance you have improved.

Explain in readme that first user is superuser... include username and password for superadmin and admin account.

Requirements

1. You should develop the front-end of your application using HTML5, JavaScript, and CSS3, and the backend using NodeJS and MongoDB as database.
2. The entry to your application should be `'http://localhost:3000/'` and the entry to the admin page should be on `"http://localhost:3000/admin/"`
3. You can use any third-party libraries with proper attribution, (No paid service, or free trial).
4. You must use a version control system (either github or bitbucket) to collaborate and share the link in your report (with login information, if private).
5. Your code should work on the Google Chrome 43+ and Firefox 40+.

Deliverables

1. **a5.zip**: All project files -- do not include the node modules folder in your zip file.
2. **A readme.pdf file** which clearly states the instruction of how to run the application.
3. **A report.pdf file** that describes the following:
 - a. A section that includes the **title** of your project, your team **information** (name and CDF of each member) and a brief **description** of your project (within sharing economy space).
 - b. A section that includes the **detailed design** of your application. This section must include the following items:
 - i. **A high-level view** of your software (i.e., not MVC or N-tier, different section of your system).
 - ii. **The description of all these different sections** of your system and how they interact with one another.
 - iii. **The list of pages and UI elements** of your project.
 - c. A section that explains how you have taken care of **security vulnerability** and how you have **tested** it in your application. If you use any framework that take care of those, describe that too.
 - d. A section describing how much **performance you have improved** after applying optimization techniques.
 - e. A section that Includes a **youtube link of a three minutes video demo** of your application(provide username and password if it private).
 - f. Anything else you want to include about your project!

Your assignment should be submitted on MarkUs.

Rubric

The following rubric will be used for for marking this assignment.

Readme: /5

- clearly states the instruction of how to run the application

Creativity: /10

-is the project creative enough to distinguish it from existing site. It ok to based on some existing application. But you should make some effort to **make it different** in some way.

Functionality: / 40

Explain that this is a first of its kind webapp.

- Satisfies the information representation of the report

- Satisfies the feature and functionality specification of the report

Breakdown:

User / Profiles / Authentication (includes both in-site and 3rd party authentication) (10pt)

Rates / Comments (5pt)

Recommendation System / Search (10pt)

Social Network (10pt)

Admin (5pt)

Testing: /15

- have demonstrate proper use of testing framework.
- have tested the functionality implemented.

Architecture (3-tier, MVC, etc.): / 15

- Follows a good architecture
- functionality is kept out of views
- Good use is made of the database, hopefully through an ORM
- The REST apis are designed properly.

Performance: /15

- Demonstrate how you have tested and improved the performance of your site
- Show comparative graphs of your site before and after optimization.

Security: /15

- What preventative measure you have taken to secure your site.
- Describe how you have tested the security of your site, include evidence.

Code Quality: / 10

- The code must be reasonably well-organized, and easy to read
- This includes general code structure and comments

User Interface and UI /10

- The site looks professional

- The site is **intuitive** to use.

Documentation: / 20

- well-organized
- provides a **good explanation** for the design decisions
- if some features are missing the documentation explains how they should work
- clearly describes **how to run and use the application**
- carefully proofread and generally acceptable grammar

Video: / 5

- A project video is included
- Is able to demonstrate the project

Note: use screencast with microphone.

Hosting in a **server: /5 (Bonus)**

- Your application is hosted in a free hosting server.
- include the URL in the report.
- make your **demo based on this URL.**

Participation Marks: /10

-contributed to the development, not only just documentation. Measured through code repository.

note: participation mark will be added as an annotation in markus and will be added into the final. It is not realistic to expect everybody to contribute equally, but the difference should not be huge.

Note: If you are developing an application other than sharing economy with my prior consent, the marking scheme will be adjusted accordingly. You have to come to the next office hours to discuss the specifics.

Sample Project 1 - CommunityFund: Improve your community

Description

CommunityFund is a [crowdfunding](#) platform for raising fund and awareness on community projects. Crowdfunding is the practice of funding a project or venture by raising monetary contributions from a large number of people, typically via the Internet. CrowdFunding has three main actors: the project initiator who proposes the idea and/or project to be funded; individuals or groups who support the idea; and a platform that brings the parties together. Since, crowdfunding leverage the power of internet, anyone from anywhere in the world can fund any project. However, CommunityFund is a community based crowdfunding platform where people would fund community-based projects. So both the funder and the project initiator has to be part of a same community. A community can be based on interests or geographical locations. CrowdFunding is an example of a [sharing economy](#) paradigm, a socio-economic system built around the sharing of human and physical resources.

How it Works

CommunityFund is an online service where an individual (or a group) (from now on, the “funder”) who are interested in helping projects within their community can find interesting projects from various project initiators (from now on, the “Initiator”). Initiators, register on the system and provide details of their projects and themselves. Funder, register on the system and search for interesting projects to fund, and if found one, they can give fund to the project. The fund represents either (i) a donation, or (ii) a payment for a product, which will be shipped to the funder at a later time (after successful completion of the project). Initiators has to specify communities for the project; which can be based on location or interests. Funders also need to specify the communities they are interested in and they can only fund any projects within those communities. Once the funder finds a suitable project they can fund the project helping it reaching a specific goal set by the initiators. The community members should form a social network Once a funder or initiator joins a community she is automatically joining a social network that allows her to explore information about the community and the projects within. They can communicate with each other and know more about the project from the people around the project.

Challenges for CommunityFund

Entrepreneurs often struggle to run their project due to lack of fund. At the same time, there are thousands of people who are willing to give money for a cause they believe in. Matching these two parties in a easy and natural way is the main challenge of this project. Again, funder always want to know the impact they have made for the project and the impact of the project within the community. So, providing the ways to measure the impact of the project within the community is an additional challenge. One possible way to measure the impact is to get testimonials (or reviews, reputation systems) from the community members for the project. Students can come up with their own ways for measuring impact.

Your Project

Your team is in charge of designing and developing a fully functional web application for supporting crowdfunding platform. Here is a list of components that your system should support:

- **User Authentication:** Authentication is the process of verifying who a person is. There are a number of methods for doing this, but the most common process is a two way matching process between a public identifier (i.e. a user name or userid) and a private identifier (i.e. a password). On the internet, this is the most common and convenient mechanism.
- **User Profiling:** Each user in your system has a profile. A user profile is a collection of personal data associated to a specific user and it may include information about her identity, her reputation, the type of project she is interested in doing or funding, the professional skills and projects she is interested in, etc.
- **User Interactions:** A user can act either as Funder or an Initiator. As an Initiator she can post a project to raise fund, describe the project milestones/products (if any), and other relevant information. As a Funder she can navigate through the listed projects, give testimonials, or give fund.
- **Implicit Social Networking:** Your system should be the basis for an implicit social network. In this social network users that are within a common community at the same time are automatically considered 'friends'.
- **Reputation System:** Your system will resemble a reputation system that computes and publishes reputation scores for both (i) funders and (ii) initiators registered on the system. The reputation score of a project is based on ratings that community members provide about a specific project. The reputation score of a initiators is based on ratings that community members provide about the initiator in their social network, based on their chance to interact or collaborate with her. Ratings are typically passed using a simple rating system (like/dislike, star system, 1-10 grade, etc.). Reputation scores are representing a collection of community member's opinion and can be used by the system to provide recommendations about which project to fund and who are good funder to seek for.
- **Administrative View:** Your system should support an administrative view of the system. Administrators can see aggregate information about the projects, such as, total number of project funded, or average days to reach a fund goal, and other useful analytics.

Sample Project 2 - SynergySpace: Working Better Together

Description

SynergySpace is a coworking space rental and teaming to succeed service. Coworking is a style of work that involves a shared working environment, often an office, and independent activity. Unlike in a typical office environment, those coworking are usually not employed by the same organization. Typically it is attractive to work-at-home professionals, independent contractors, or people who travel frequently who end up working in relative isolation. Coworking is also the social gathering of a group of people who are still working independently, but who share values, and who are interested in the synergy that can happen from working with people who value working in the same place alongside each other. Coworking is an example of a sharing economy paradigm, a socio-economic system built around the sharing of human and physical resources.

How it Works

SynergySpace is an online service where individuals that own a space (from now on, the “Owner”) connect with individuals that are looking for temporary working space (from now on, the “Tenant”). Owners, register on the system and list the space they would like to rent. Tenants, register on the system and search for space to rent. Owners provide information about the space and Tenants provide information about themselves. After a space is found, an arrangement is made that includes information about the details of the lease. Once the Tenant physically joins a shared space she is automatically joining a social network that allows her to explore information about other Tenants currently in the shared space. The social network can then be seen as an enabler of interacting with other people with similar interests. Eventually Tenants have the chance to meet potential collaborators and build up a team to work on a project towards synergies. Currently, there are online coworking services where an individual can rent a space and there are online services that can enable finding partners to work on a project. However, these services are not integrated well and are not enabled through physically sharing a space at real-time. A service that enables this interaction would be highly desirable as it would give extra flexibility on finding a working space, and at the same time would enable more meaningful synergies between interested individuals.

Challenges for SynergySpace

Freelance professionals can struggle to find affordable working space in a city they reside or are visiting. At the same time, space owners are looking for alternative ways to lease their space. Another concern for individuals is that they are typically looking for new opportunities to join a team and collaborate on a project. While it is easy to find collaborators online, many individuals would prefer to work with professionals that they can physically interact. Another problem for coworking is the reliability of the informal arrangements made between the parties involved. Several internet schemes are trying to address these concerns by introducing reputation systems, enabling more informed decisions to be made about reliable and unreliable professionals.

Your Project

Your team is in charge of designing and developing a fully functional web application for supporting the rental of coworking space and the teaming of professionals online. Here is a list of components that your system should support:

- **User Authentication:** Authentication is the process of verifying who a person is. There are a number of methods for doing this, but the most common process is a two way matching process between a public identifier (i.e. a user name or userid) and a private identifier (i.e. a password). On the internet, this is the most common and convenient mechanism.
- **User Profiling:** Each user in your system has a profile. A user profile is a collection of personal data associated to a specific user and it may include information about her identity, her reputation, the space she is interested to list for lease or is interested to rent, the professional skills and projects she is interested in, etc.
- **User Interactions:** A user can act either as Owner or a Tenant. As an Owner she can list a new space for lease, arrange the lease agreement, rate individuals that have leased the space. As a Tenant she can navigate through the available rental spaces, express interest to lease a space, rate a space.
- **Implicit Social Networking:** Your system should be the basis for an implicit social network. In this social network users that are sharing the same space at the same time are automatically considered 'friends'.
- **Reputation System:** Your system will resemble a reputation system that computes and publishes reputation scores for both (i) coworking spaces and (ii) Tenants registered on the system. The reputation score of a coworking space is based on ratings that Tenants provide about how satisfied they were with the space (and the details of the arrangement). The reputation score of a Tenant is based on ratings that Tenants provide about other Tenants in their social network, based on their chance to interact or collaborate with them. Ratings are typically passed using a simple rating system (like/dislike, star system, 1-10 grade, etc.). Reputation scores are representing a collection of Tenants' opinion and can be used by the system to provide recommendations about which working space to lease and which people to collaborate with once you are there.
- **Administrative View:** Your system should support an administrative view of the system. Administrators can set global variables used in the system, such as, the type of space that can be leased and can see aggregate information about the system's performance, such as, total number of teams built up, or average distance that users had to travel to find a coworking space, and other useful analytics.