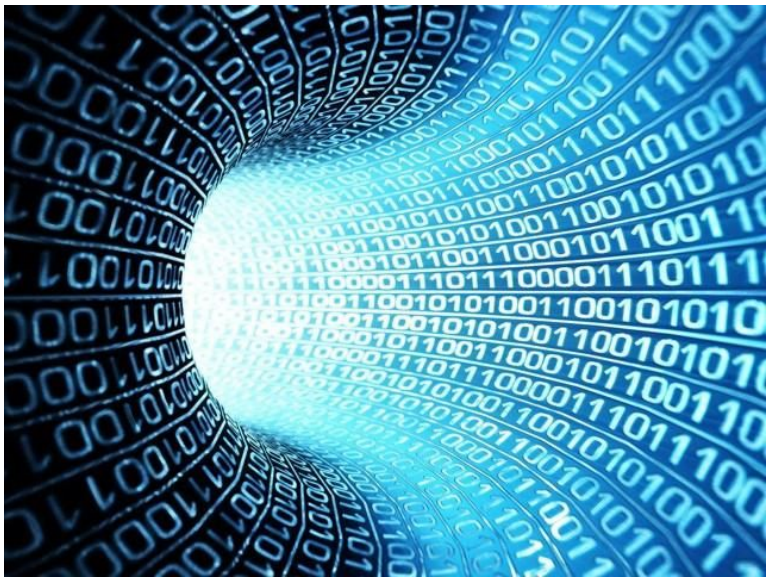
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
Campus Monterrey

Departamento de tecnologías computacionales
TC2026. Desarrollo de Aplicaciones Web
Martes y Viernes de 7:00 – 8:30

Proyecto EmployerDB



Profesor titular:
Ing. Alfredo Salazar

Elaborado por:

Alumno: Daniel Castro Juárez
Matricula: A01089938

Alumno: Marving Robles Angeles
Matrícula: A01651377

Monterrey, N.L.; 06 de Noviembre del 2019

I. Project description

a. Decide why you need a web application.

Companies have a high necessity to keep track employees inside projects. Normally, there is no control in the companies related to this topic. This will be helpful in many ways. Someone inside the company or the team leaders could search for people inside the company with certain skill set, formation, education, qualifications to be part of their team in their project.

The general idea is to keep a record of the projects and people working in them in a clean, and easy way for someone to select people with the best match for their team. This can be done by:

- Publish projects with their description and important information related to what the project is about.
- Publish people with all the information related to their skillset
- Look up information about a project registered
- Look up information about a person registered

With all the information inside the Database in the long term we can use machine learning to see certain patterns and analysis.

b. Scope

Any company and/or anyone with the necessity to keep track of their projects and teams can use this application. The user doesn't need to do anything, he just needs to create a user and password in the web browser of their preference.

c. Vision

The website will help the companies or anyone to track all their projects and employees (team members). It will offer meaningful data for future projects, in which people can see what projects their team members have been working on.

d. Objective

- The user will be able to track employees/team members history
- The user will be able to track projects history
- The users can post information about the project
- The users can post information about the people registered
- Update / Delete a Project Data
- Update / Delete a Person Data

II. Requirements

a. Functionality and usability

The system allows small business owners to have control of project members, to know what was done through publications, for this the system must have functions such as:

- Log in / out
- Create Profiles
- Create projects
- Add members to projects
- Project members can make publications

To maintain the usability of the website, a clean and minimalist design will be made, with which it is easy to interact, the navigation menu will be present in all windows so browsing through the website will not be a problem.

b. Style and layout

The fonts for titles will be selected from Google Fonts

i. Login and Registration page

1. In the top section of the page will be displayed the logo along with a navigation menu.
2. On the center of the page the user will be able to register or to login, on the bottom side will be displayed information about the web site.

ii. Home page

1. At the top we will have our banner to search in the side, and a navigation menu, on the top left side the logo will be displayed all the time.
2. in the main part of the page will be information about groups, projects, and general posts.

iii. Groups page

1. The same banner will be present, but in the main part will only be posts visible for the members of the groups.

c. Functional requirements

i. User Requirements

ID	FR01
Use case scenario	Login
Functionality	The user enters in the system
Precondition	Page shows a mail field, a password field and a login button
Basic flow	1. The user types mail and password

	2. The user clicks on login button
Alternative flow	If the user is not registered he/she might register, else won't be able to see anything in the site.
Postcondition	The user enters the main site
Exception flow	If any connection fails, an error message will be displayed.

ID	FR02
Use case scenario	Logout
Functionality	The user logs out of the site
Precondition	The page has a visible logout button
Basic flow	1. The user clicks on Logout
Alternative flow	The user exits the browser before logging out
Postcondition	The session ends and user is redirected to login page
Exception flow	If any connection fails, an error message will be displayed.

ID	FR03
Use case scenario	Create a project
Functionality	The user creates a project to manage information and members.
Precondition	The user clicks in “create group”
Basic flow	1. The user completes the fields: a. Title

	b. Description c. Creates tasks 2. The user clicks on create group 3. The manager makes a presentation post
Alternative flow	If the user writes a group title that is repeated, it will display error.
Postcondition	The project is created
Exception flow	If any connection fails, an error message will be displayed.

ID	FR04
Use case scenario	Make a post
Functionality	The user can make a post and everyone will be able to see it
Precondition	The user is in posts area
Basic flow	1. The user clicks on make a post 2. The user writes header, and content 3. The user clicks on Post button
Alternative flow	If the user leaves the browser before posting, it will be lost
Postcondition	The post is visible in the projects main page
Exception flow	If any connection fails, an error message will be displayed.

ID	FR05
Use case scenario	Look for users
Functionality	The user can look for other users in the site

Precondition	The user is in the look person tab
Basic flow	1. The user looks for the name of the user in the list 2. The user writes the id of the user in the id field 3. The user clicks in search button
Alternative flow	The user is not found
Postcondition	The user information is displayed.
Exception flow	If any connection fails, an error message will be displayed.

ID	FR06
Use case scenario	Look for project
Functionality	The user needs to see information about a project.
Precondition	The user is in the look project page
Basic flow	1. The user looks for the project name in a list 2. The user writes the id of the project in the if field 3. The user clicks in search button
Alternative flow	The project is not found
Postcondition	The project information is displayed
Exception flow	If any connection fails, an error message will be displayed.

ID	FR07
Use case scenario	Update person
Functionality	The user can update the information about people in the database
Precondition	The user is in the search person page

	and knows the id of the person
Basic flow	1. The user types the id in the id field 2. The user clicks in get button 3. The user types the new information in the fields 4. The user clicks in update button
Alternative flow	The user is not found. If the user leaves the browser before clicking in update, changes will not be applied.
Postcondition	The person information is displayed updated
Exception flow	If any connection fails, an error message will be displayed.

ID	FR08
Use case scenario	Update project
Functionality	The user can update information about projects
Precondition	The user is in the update page and knows the project id
Basic flow	1. The user types the id in the field 2. The user clicks the button get 3. The user types the new information 4. The user clicks in update button
Alternative flow	The project is not found. If the user leaves the browser before clicking in update, changes will not be applied.
Postcondition	The project information is displayed
Exception flow	If any connection fails, an error message will be displayed.

ID	FR09
Use case scenario	Delete person
Functionality	The user can delete persons from database
Precondition	The user is in the search project page and knows his/her id
Basic flow	1. The user writes the id in the if field 2. The user clicks in search button 3. The user clicks on Delete
Alternative flow	The user exits the page before clicking on delete
Postcondition	The user information is removed from database
Exception flow	If any connection fails, an error message will be displayed.

ID	FR010
Use case scenario	Delete project
Functionality	The user can delete his/her projects from database
Precondition	The user is in the search project page and knows the project id
Basic flow	1. The user makes a search with id of the project 2. The user clicks on Delete button
Alternative flow	The user exits the page before clicking on delete
Postcondition	The project information is removed from database

Exception flow	If any connection fails, an error message will be displayed.

d. Non-functional requirements

i. User Requirements

ii. System Requirements

III. Specifications

a. Functional Specifications

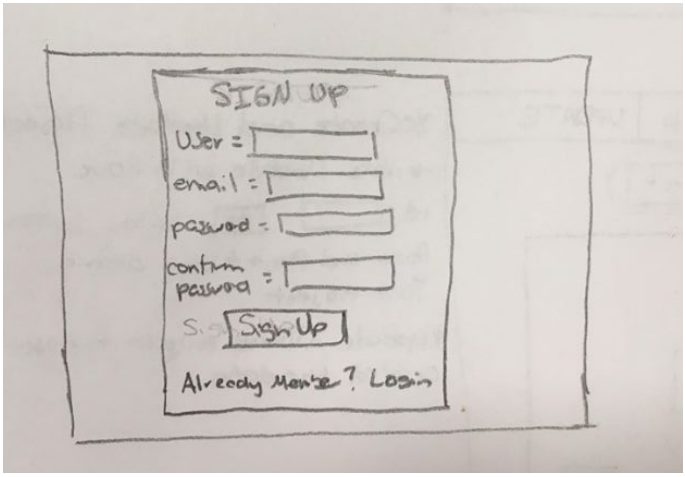
In this section are defined the necessary functions to achieve the functional requirements described with the parameters.

- Login(userMail: Mail, password: String)
  - Makes an Ajax call to the backend to validate data and make a match
- Logout()
  - Ajax call to terminate session and redirect to login page
- getAllUsers()
  - Ajax call to get a list of all the users registered in the database
- getAllProjects()
  - Ajax call to get a list of all the projects registered in the database
- searchPerson(Id: String)
  - Ajax call to make a match and retrieve a person's information
- SearchProject(Id:String)
  - Ajax call to make a match and retrieve a project's information
- CreateProject(name:String, size: Int, description: String, manager: Object)
  - API call to backend to create a project with all the fields completed. The manager will be selected from a combobox
- CreateUser(name:String, age: Int, degree: String, birthday: Date, skills: String, mail: Mail)
  - API call to backend to create a new person profile, all fields have to be completed.
- UpdatePerson(id: String, name:String, age: Int, degree: String, birthday: Date, skills: String, mail: Mail)
  - Ajax call to make a match and retrieve information, the information will be changed for the new fields.

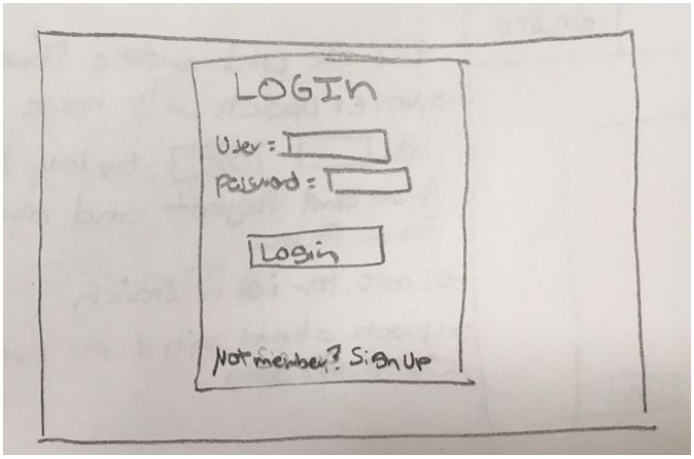
b. Design Specifications

The following are the screens of the application so we have a better understanding of what we want to build.

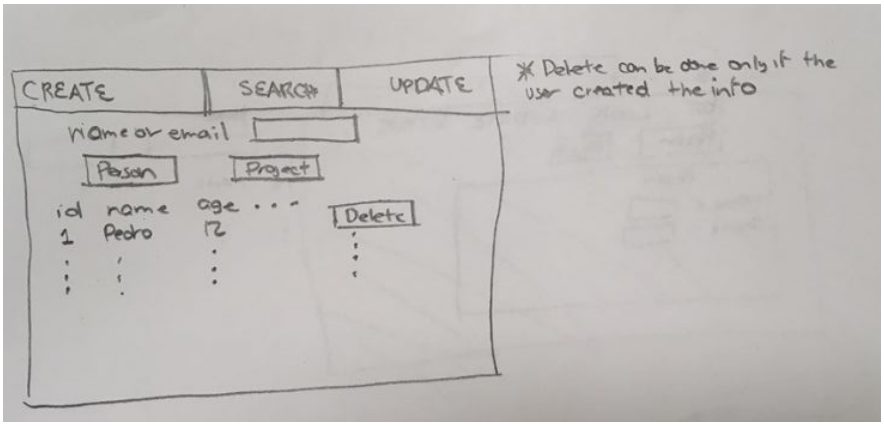
Sign up



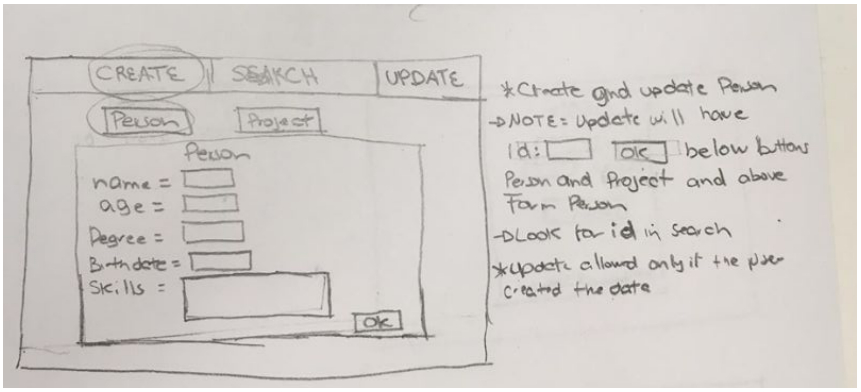
Login



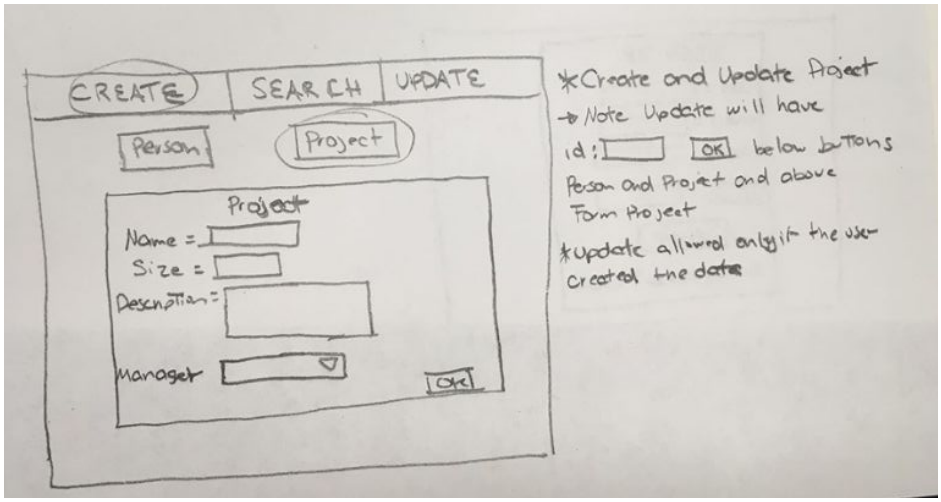
Search a post



Create a person



Create a project



c. Technical Specifications

The following are the technical tools that will be used in the project:

- Front-End
  - HTML
  - CSS
  - Javascript
  - JQuery
- Back-End
  - Express
  - Mongo
  - Mongoose
  - Postman

IV. System Architecture

a. Software architectural pattern

i. Model - View - Controller

b. **Development platform**
    i. MongoDB, Postman, Sublime
c. **Components**
    i. Front-end
    ii. Back-end

I.a = Decide why you need a web application.