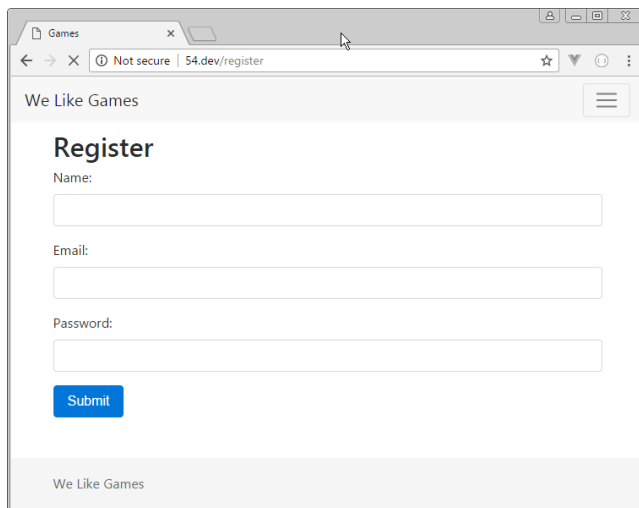


The purpose of the application is to simulate a system that allows you to manage tasks and resources, and that implements communication through a chat. The Stack MEAN.js technology will be used, that is, MongoDB as a non-relational database, Express, Angular and Node

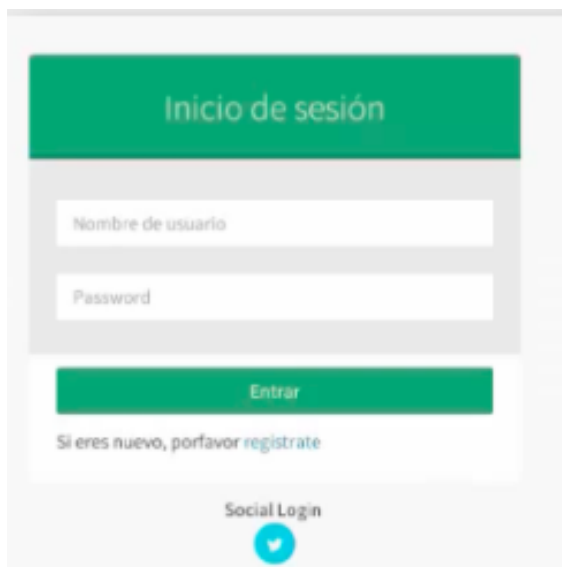
1. Develop an application using MEAN.js Stack that allows the management of tasks and resources of a particular company. For this, the application must have the following characteristics and functionalities: Develop a platform registration system for users. For this, an interface such as the following must be developed:



The screenshot shows a web browser window with the address bar displaying '54.dev/register'. The page title is 'We Like Games'. The main content area features a registration form with the following elements:

- Header:** 'We Like Games' with a hamburger menu icon on the right.
- Title:** 'Register' in bold.
- Form Fields:** Three input fields labeled 'Name:', 'Email:', and 'Password:'.
- Submit Button:** A blue button labeled 'Submit'.
- Footer:** 'We Like Games' at the bottom.

2. The application must allow the login through registered users, using the username and password. Similarly, you must be able to log in through the social network Twitter. An interface for the login must be developed, as shown in the following image:



The screenshot shows a login interface with the following elements:

- Header:** A green bar with the text 'Inicio de sesión' in white.
- Form Fields:** Two input fields labeled 'Nombre de usuario' and 'Password'.
- Login Button:** A green button labeled 'Entrar'.
- Registration Link:** A link that says 'Si eres nuevo, porfavor [regístrate](#)'.
- Social Login:** A section labeled 'Social Login' with a Twitter icon.

3. A Tasks module must be developed. In this module the user can manage tasks, from creating them to being able to finish them. An interface that allows the management of tasks should be developed, and a component or text box that allows entering the name of the task should be developed. Next, two button type components must be developed, one to create the task and another to save the changes made to it. Task must have date and completed tag when its finished.

The following images show how the interface should look:

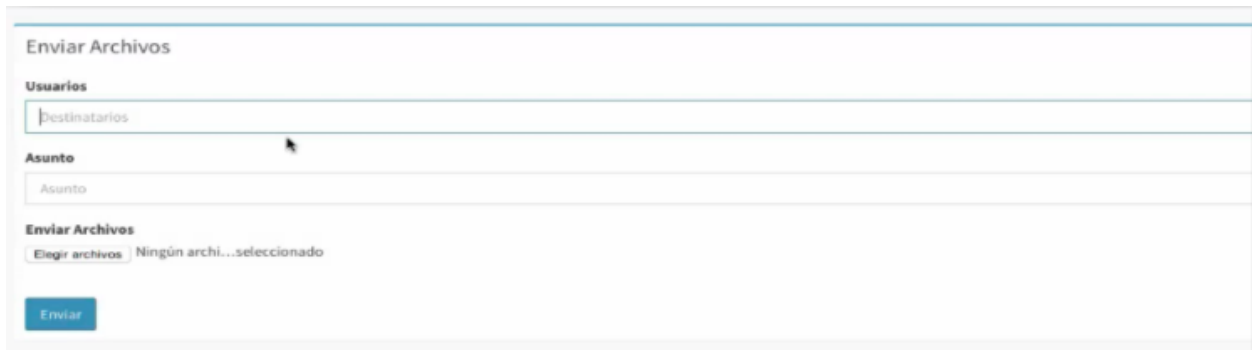
The screenshot shows a web application interface for the 'Tareas' module. At the top, there is a blue header bar with a user profile icon and the name 'Gustavo De La Vega'. Below the header, the page title 'Tareas Módulo' is displayed. The main content area is titled 'To Do List' and contains a form with a text input field labeled 'Descripcion', a blue 'Clear' button, and a list item 'Primera Tarea' with a checkbox. A 'Guardar' button is located at the bottom right of the form.

The screenshot shows the same 'Tareas' module interface, but with an additional table at the bottom titled 'Historial de Tareas Finalizadas'. The table has three columns: 'Descripcion', 'Fecha', and 'Estado'. It contains three rows of data, each with a task description, the date '2015-05-05', and a green 'Finalizada' status tag.

Descripcion	Fecha	Estado
Realizar La Actividad de esta Leccion	2015-05-05	Finalizada
Primera Tarea	2015-05-05	Finalizada
Realizar La ACTIVIDAD DEL QUIZ	2015-05-05	Finalizada

4. A Resources module should be developed, and this should allow the user to send resources to another user. In the same way it will be possible to visualize the resources sent and received. A button type component must be developed, which when pressed

must show a form where the data of the resource submission will be digitized, as shown in the following image:



The screenshot shows a web form titled "Enviar Archivos". It has three main sections: "Usuarios" with a text input field containing "destinatarios"; "Asunto" with a text input field containing "Asunto"; and "Enviar Archivos" with a button labeled "Elegir archivos" and the text "Ningún archi...seleccionado". At the bottom is a blue button labeled "Enviar".

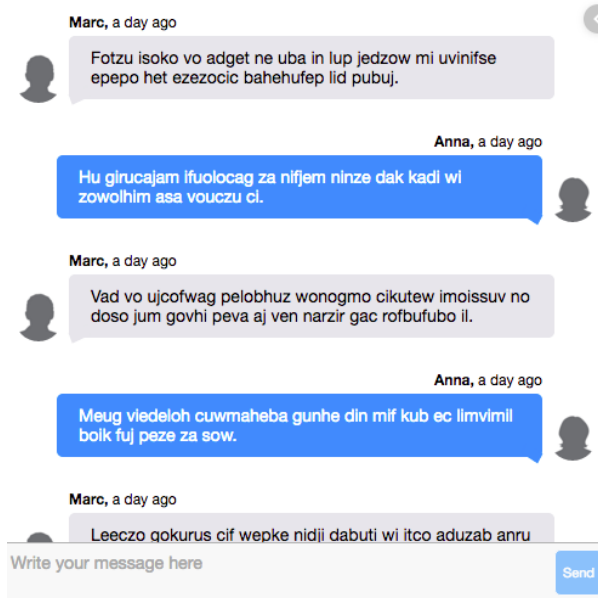
The Resources module can show the resources received and the resources sent, as shown in the following image:



The screenshot shows the "Recursos" module interface. On the left is a sidebar with "Inbox" and a blue button "Enviar Recursos". Below it are "FOLDERS" with "Recibidos" and "Enviados". The main area shows a list of resources:

From	Subject	Time
Gustavo De La Vega	Envío una Imagen	12:30 PM
Gustavo De La Vega	Archivo	12:30 PM

5. A Chat module must be developed. Here users can write and receive messages in real time individually and in general, using Socket.IO. An interface must be developed showing the connected users and the option of general communication with all users, as well as the following images:



6. A Dashboard or Timeline module must be created. Here the actions created by the other users and their own will appear, being these actions finalize tasks and share resources. It should work in real time, using Socket.IO. The following image is a sample of what the interface should look like:



I should be able to run this app in a local server, both the backend and frontend.