



Freelance System

Problem Definition:

If you're looking for a flexible schedule in which you can make your own hours, work from home during this pandemic, then a freelance job might be for you. A lot of work can be done remotely these days, so your dreams of working from your couch can come true by **developing a freelance web application** to manage all of these.

Freelance web applications are platforms, where both people looking for work and employers post their offers.

Actors (Three actors):

- Admin
- Freelancer
- Clients

Layouts (Three layouts):

1. We need a web application (ASP.Net MVC) which handles and integrates the communication between freelancers (Ex: developer) and the clients (person that offers a jobs and will pay to the freelancer in order to do the client's job)
2. The web application includes 3 layouts
 - First layout: the freelance layout and we will call it the **wall Layout** (all the jobs posts exist there)
 - Second layout: The client layout and we will call it the **factory Layout** (as the client writes and edits jobs posts in order to hire a freelancer)
 - Third Layout: The admin layout and we will call it the **dashboard Layout** (where admin can control clients and their posts and also control freelancers)

Actors Roles:

1. **Admin:**
 - Control all users (Freelancers and Clients): Admin can Add, Remove Users

- Control the **wall** page: Admin can Add, Remove and Update job posts in the wall page which are created by the client
- Admin pages (in **dashboard layout**) **include**:
 - a. Admin information (profile page): photo, first name, last name, email, phone number and user role (admin).
 - b. All the users that using the system so that the admin can control them.
 - c. All the posts that created by the **clients** so that the admin can control them.
 - d. If the Admin remove a post from his profile it reflects directly to the **wall** and be removed.
 - e. Admin Accept or Refuse Posts written by the Clients before adding it to the **Wall**.
 - f. The admin must login first, to operate his job

2. Client:

- Write job posts that will show in the **wall** page
- Client pages (in **factory layout**) **include**:
 - a) Client information (profile): photo, user name, first name, last name, email, phone number.
 - b) Client can change (update) his information
 - c) Password **Not Showed** but we need a button to allow Client to Change his password
 - d) A Form that the Client will use to write job posts and add them to the Wall.
 - e) Post contains: Client name, Job Type (fixed or hourly), Job Budget, post creation date, Job description, Number of proposal submitted.
 - f) Admin Accept or Refuse Posts written by the Clients before adding it to the **Wall**.
 - g) History that includes all job Posts that have been added before by this client in the Wall (mention date in each Post)
 - h) Client can manage his Posts (Add, Edit, Delete, Search).
 - i) Client Accept or Refuse freelancers' proposals for a certain job. If the client accepts the proposal, the job post will be removed from the **wall** page.
 - j) The Client must login first, to operate his job

3. Freelancer

- Login to apply for a job
- View (read only) all job posts from different clients (no need to login)
- Freelancer pages (in **wall layout**) **include**:
 - a) All Posts that have been added by Clients from their Profiles
 - b) Each post (created by Client) has: Client name, Job Type (fixed or hourly), Job Budget, post creation date, Job description, Number of proposal submitted.
 - c) Freelancer can search for a job by Title, Date or Client name.
 - d) Freelancer can save a particular post in saved page (to read later)
 - e) Freelancer can rate the job post (from 1 to 5 stars)
 - f) User can apply to a job by sending a proposal to the client.

Important Notes:

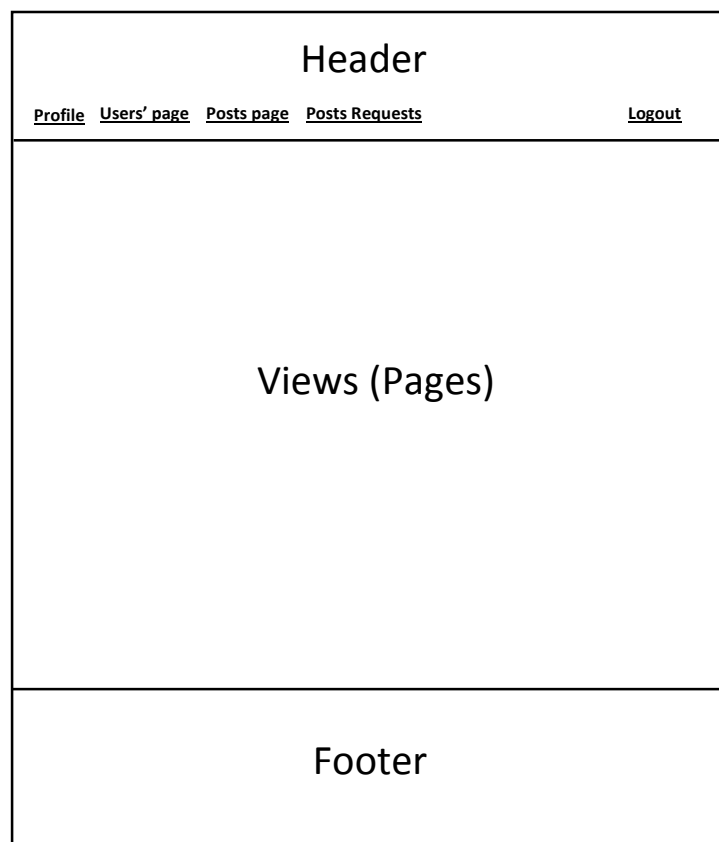
- You need to think and **design your project relational database** (tables, tables' columns and tables' relationships) and **Print the Database Schema**
- You have three layouts (Wall – Factory - Dashboard) and Many Views (Pages).
- Views Folder should include:
Shared folder contains (WallLayout.cshtml, FactoryLayout.cshtml, DashboardLayout.cshtml)
- Create two Popups (modal):
 - a. First Popup: For Login
 - b. Second Popup: For Registration.
 - The Login popup appears when a button in the Wall layout is clicked.
 - The Registration popup appears when a button in the Wall layout is clicked.
 - If the freelancer or the client doesn't have an account then he will register.
 - Registration Information: first name, last name, username, email, phone number, photo, User role (freelancer or client). **The role can't be updated later**
 - You can add admin in the database manually and then you can login.
- Your app is responsive (**Bootstrap is required for Grid system**)

Layouts:

1- Dashboard Layout (Admin)

- Include these parts (as mentioned above):
 - a. Admin profile page
 - b. Users page (add or delete users (freelancers and clients))
 - c. Posts page (delete or update posts which is already in the wall)
 - d. Posts requests (accept or refuse to add new posts (created by the clients) in the wall)

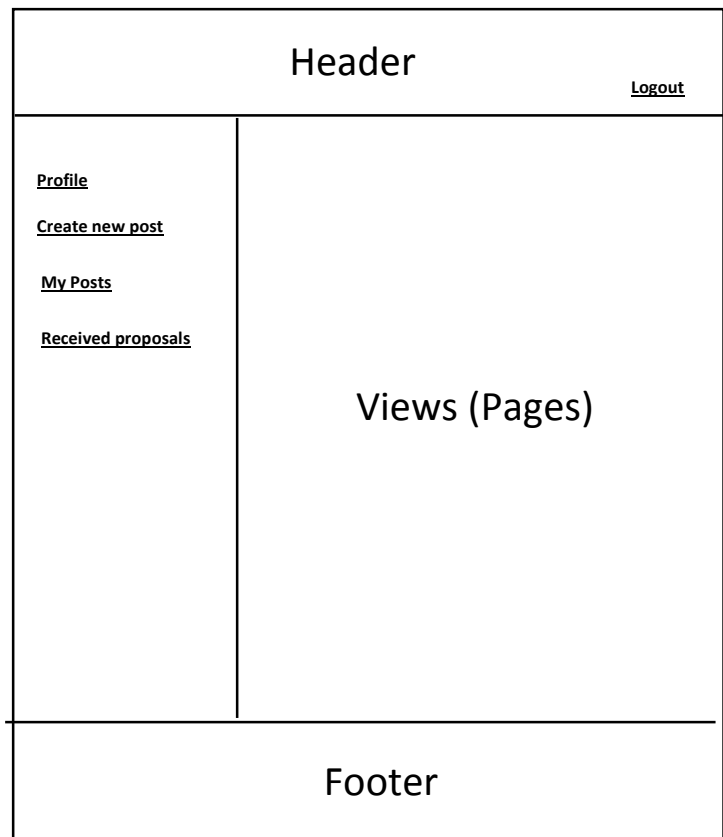
Dashboard Layout:



2- Factory Layout (Client)

- Include these pages (as mentioned above):
 - a. Client profile page (profile information is changeable except the role)
 - b. Create job post page (to write job posts)
 - c. History page (contains posts created by this client and can delete and edit and search these posts)
 - d. Received proposals (accept or refuse freelancers request for a certain job)

Factory Layout:



3- Wall Layout (Freelancer)

- Include these pages (as mentioned above):
 - a. The wall page: which contains all the job posts
You can view this page without logging (read only). Freelancer functionality (as mentioned above) will be available after logging in.
 - b. The save page: freelancer saves job posts in it to read it later.
 - c. freelancer can search for a job by Title, Date or Client name (**search in the layout**)
 - d. Login and register forms are popups

Wall Layout:

Header	
Filter: <input type="text"/>	Register Login
Footer	

Best Wishes