



|  |
| --- |
| Create By: Mohammad Alaa Aldein  Project for Orange Academy |

Abstract

Jobi is a site specialized in jobs, it is directed to all employers who need employees, jobs seekers and businesses owners who want to market theirs businesses, the site will consist of several pages and will contain a content management system as well and will be built on React.

In this site, we used the Agile methodology and took several steps (chapter 1). I identified the challenges and the solutions. drew up the project construction timeline. Then, at (chapter 2), we collected the requirements for building the site and were collected by a questionnaire and then we identified the functional and non-functional requirements. Then we use all requirements to build use case and sequence diagram, and in (chapter 3) and took Data and we have created data then founded databases and tables of data and designed Screen to the site and we started implementing the project and used the language of Javascript to build the site at (chapter 4) identified future business and the most important results that got it.

# Chapter 1: Introduction

## Introduction

Jobi is a site specialized in jobs, it is directed to all employers who need employees, jobs seekers and businesses owners who want to market theirs businesses.

Employers offer jobs applications, jobs seekers search for jobs, explore it and apply for it.

And businesses owners market theirs businesses as want

## System Goals

The goal of building the site is to:

1. Offer jobs applications from employers
2. Able to search and explore jobs by jobs seekers
3. Businesses owners market theirs businesses

## Feasibility Study

### Challenges

1. Plenty of fake job offers.
2. A lot of competitive sites.

### Proposed Methods

1. Facilitate to jobs seekers to find jobs easily and employers to offer jobs.
2. Get real jobs offers.
3. Market businesses from owners as want

## The Method Adopted in The Design of The System

1. Object-oriented design includes two main stages, namely, system.
2. System Design: System Design: In this stage, the complete architecture of the desired system is designed. The system is conceived as a set of interacting subsystems that in turn is composed of a hierarchy of interacting objects, grouped into classes.
3. System design is done according to both the system analysis model and the proposed system architecture.

### Analyze & Requirement

In the first mission I analyze and gather information to gather requirements

Chapter 2: Requirement

Analysis

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mold user expectations to fit the requirements.

*User Requirement*

This field of indoor and outdoor maintenance and all services with just one click with entering the problem (The main features:)

**A-Front Web Site**:

1. Login and register page for new users

2. Home page contain slider

3. Jobs page contain jobs and job page to view job details

4. Jobs application contain jobs applications and job application page to view job application details

5. Companies page contain companies and company page to view company details

6. Forms pages to add jobs and jobs applications

**B-Back Web site:**

1. Admin Dashboard to control of web site

2. Login and Register pages

3. User Dashboard to control user information and application they submit

4. Search page to search for anything

#### Functional requirement

1. log in / register page
2. Dashboard page to control user information
3. Home page contain slider
4. **Jobs, Jobs Applications** to show jobs and jobs applications
5. Job, Job Application to show job and job application details
6. Companies page to show companies

#### Non-Functional Requirement

1. External Requirement

* The web site has Admin to check, edit and delete jobs and companies.

1. Organizational Requirement
2. Require any OS (such as Windows (XP, 7, 8.1, 10) Linux and Mac). Etc.
3. Web browser (chrome, Firefox, Opera, Explorer).
4. Connected to the Internet.
5. To Program Web pages we use the Mern-Stack React and connect it with MongoDB, Web pages CSS, Html.

.

#### System requirement

1. The web site must be compatible with pc’s and mobiles devises.
   * To achieve all these functions a computer or mobile devise must be available and connected to the internet.
   * The version or model of the device does not matter, but defiantly needs at least a web browser.
2. Programming languages used

* HTML (Hyper Text Markup Language) is the most basic building block of the Web.
* CSS is a language that describes the style of an HTML document.
* JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions.

#### Use Case Description

1. Actors:

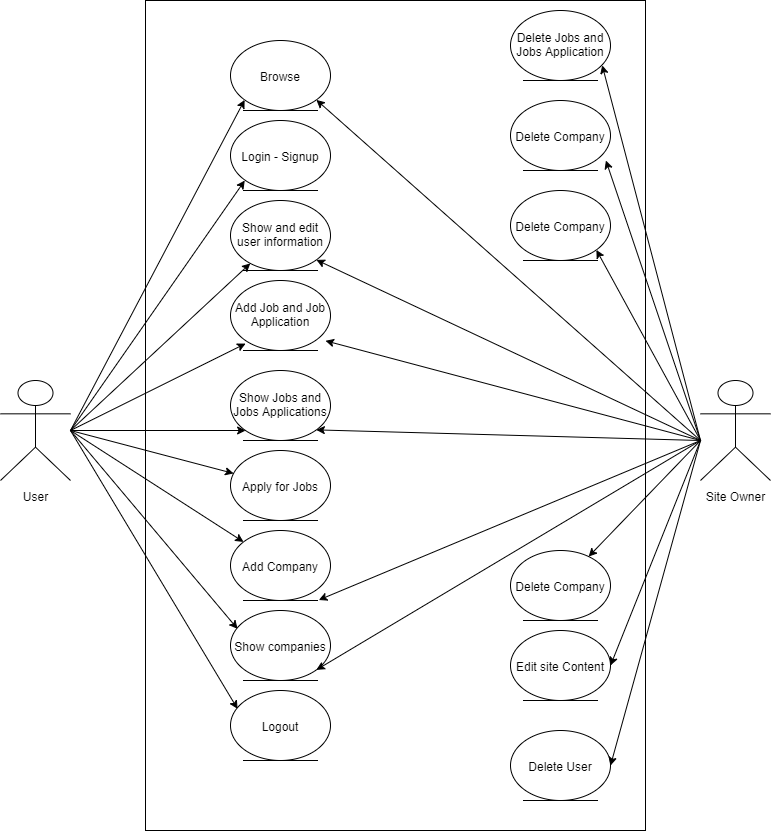
a. Human actors: Admin, Editors, Bloggers.

|  |  |
| --- | --- |
| **Use case** **Number** | **1** |
| **Use case name** | Browsing |
| **Participation actor** | 1. User 2. Admin |
| **Pre-condition** | Internet connection |
| **Flow of events** | 1. Open web site 2. Browse the web site |
| **Post-condition** | Retrieving, presenting and traversing information resources |
| **Quality**  **Requirements** | Speed site browsing |

|  |  |
| --- | --- |
| **Use case** **Number** | **2** |
| **Use case name** | Sign up for users |
| **Participation actor** | 1. Users |
| **Pre-condition** | Internet connection, e-mail |
| **Flow of events** | 1. open website 2. click on sign up 3. insert information 4. check information 5. send to server |
| **Post-condition** | Fill more information in user dashboard |
| **Quality**  **Requirements** | 1. If the user entered an incorrect Email or password. 2. Show error messages: sign up fails, Please try again. |

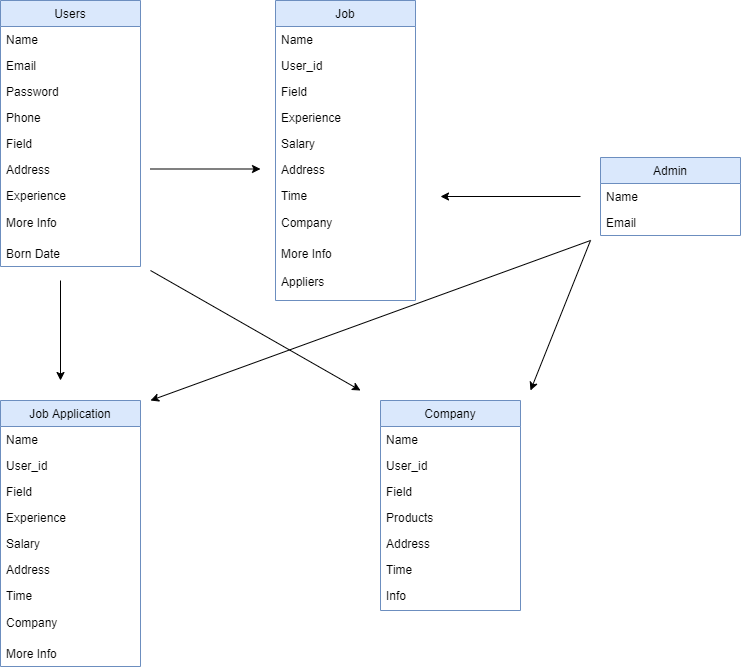
|  |  |
| --- | --- |
| **Use case** **Number** | **3** |
| **Use case name** | Add Job |
| **Participation actor** | 1. User as job employer |
| **Pre-condition** | Internet access and job details |
| **Flow of events** | 1. Go to job page 2. Click on Add Job 3. Insert all data 4. Click Add Job to submit form data |
| **Post-condition** | ------------------- |
| **Quality**  **Requirements** | User can apply for jobs or add jobs after signup |

##### Use Case Diagram



Chapter 3: Design

Class diagram



Database Tables

Database: Table Users

|  |  |
| --- | --- |
| Data Name | Type |
| Id | big Increments |
| Name | String |
| Password | String |
| Email | String |
| Field | String |
| Address | String |
| Experience | Number |
| Info | String |
| Born\_Date | Date |

Database: Table Jobs

|  |  |
| --- | --- |
| Data Name | Type |
| Id | big Increments |
| Name | String |
| Field | String |
| Company | String |
| Experience | Number |
| Salary | Number |
| Time | String |
| Address | String |
| Info | String |
| User\_id | String |
| Applied | String |

Data Base: Table Jobs Applications

|  |  |
| --- | --- |
| Data Name | Type |
| Id | big Increments |
| Name | String |
| Field | String |
| Experience | Number |
| Salary | String |
| Time | String |
| Address | String |
| Info | String |
| User\_id | String |

Data Base: Table Companies

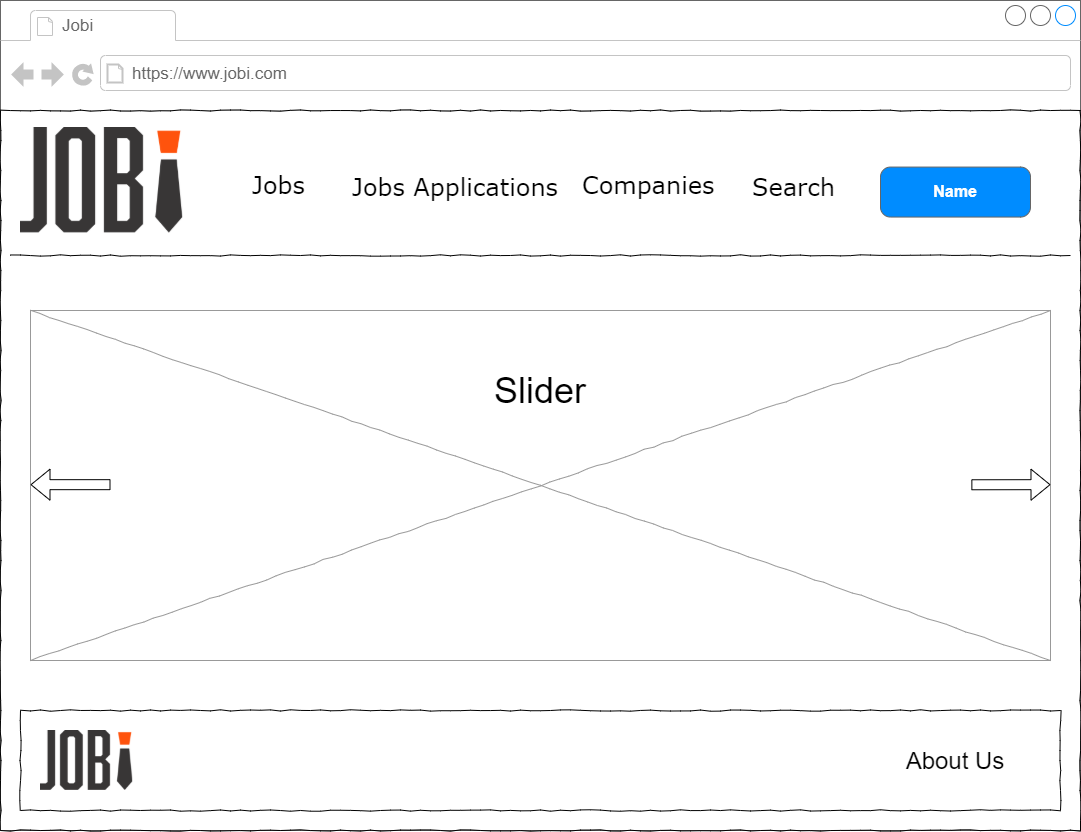
|  |  |
| --- | --- |
| Data Name | Type |
| Id | big Increments |
| Name | String |
| Field | String |
| Products | Array |
| Address | String |
| Time | String |
| Info | String |

Mock-ups

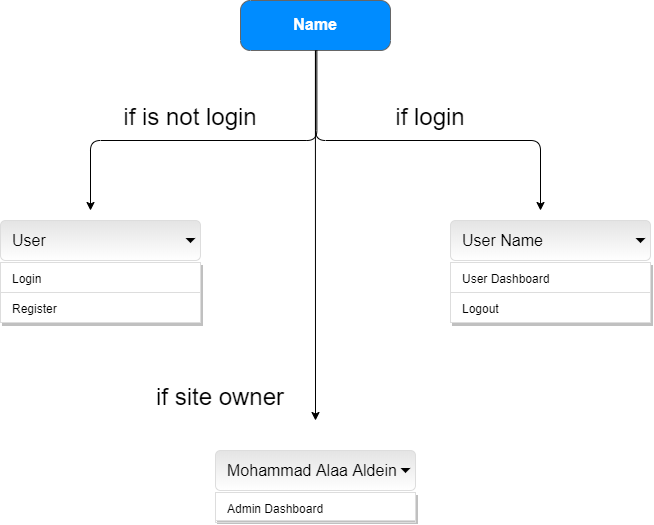
Is a design work for website design screen?

Website Mock-ups

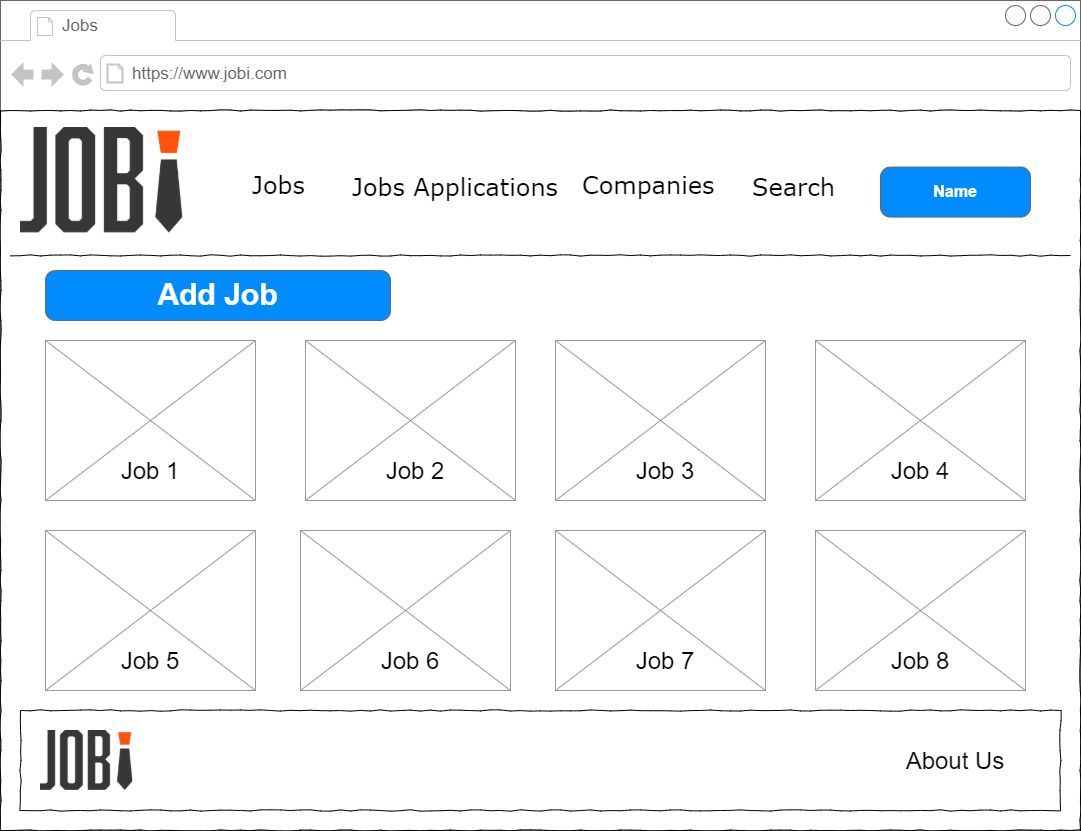
*Home Page*

**

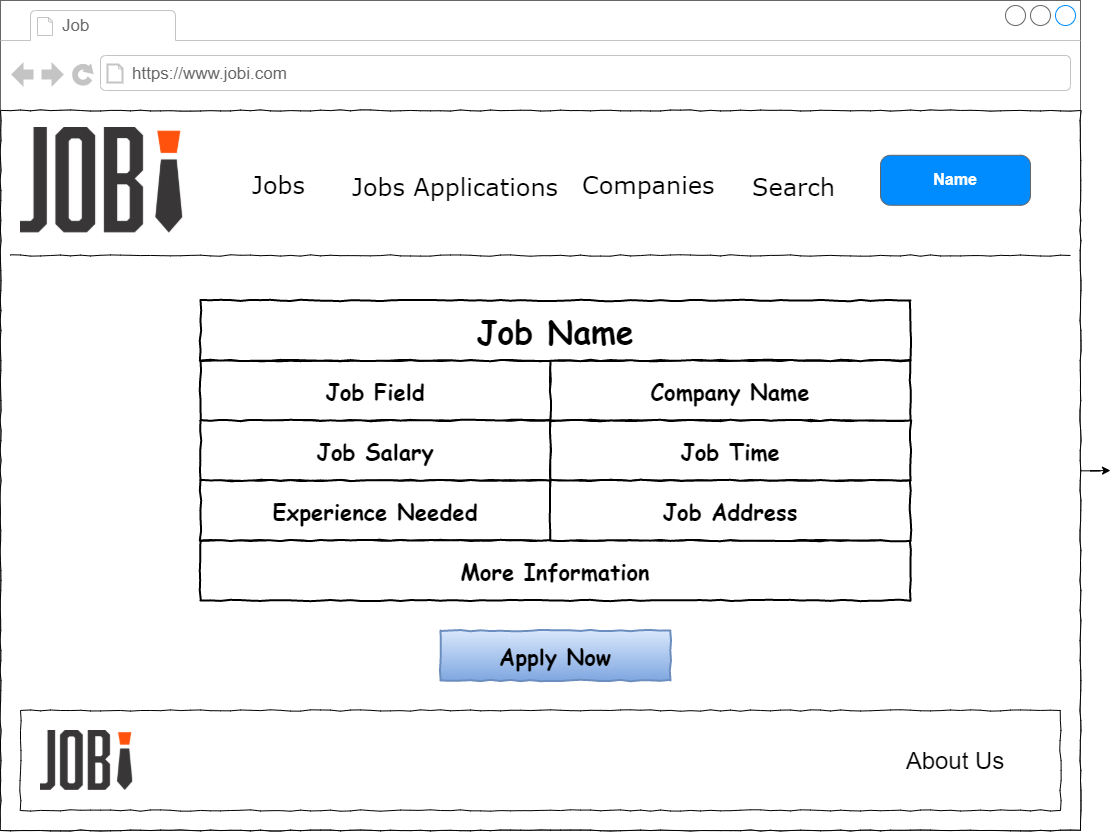
*When Press - User - Button*



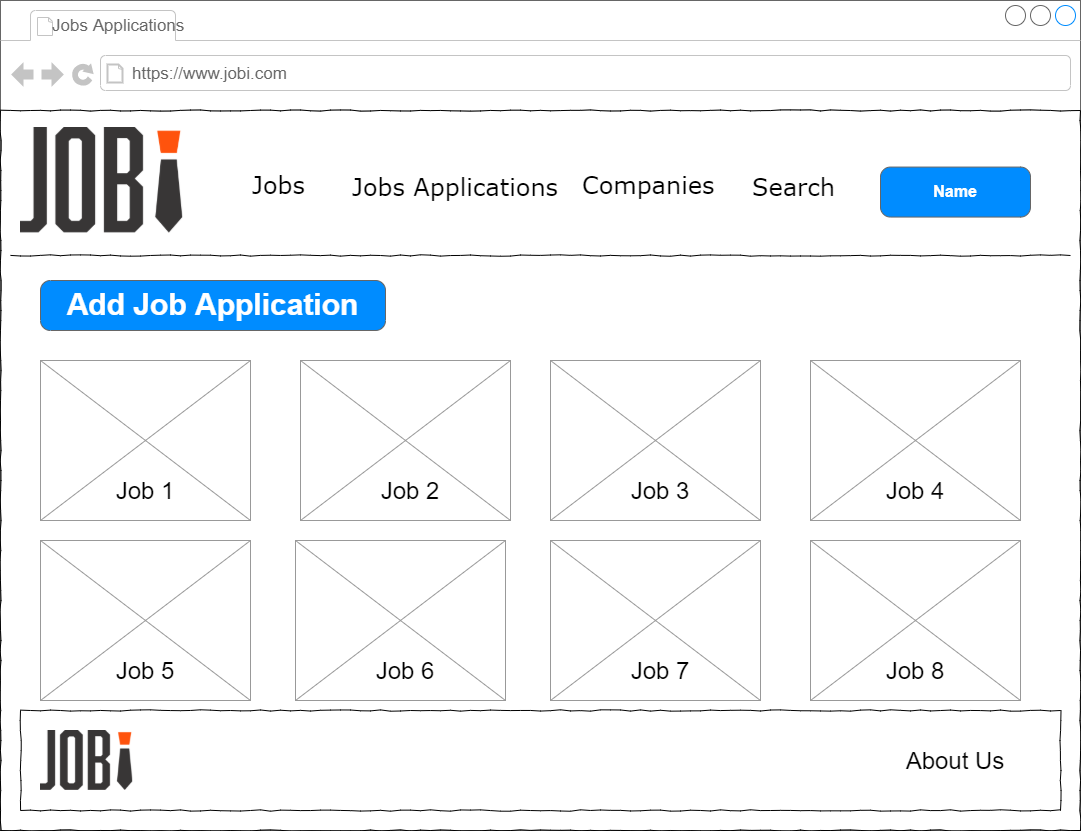
*Jobs Page*



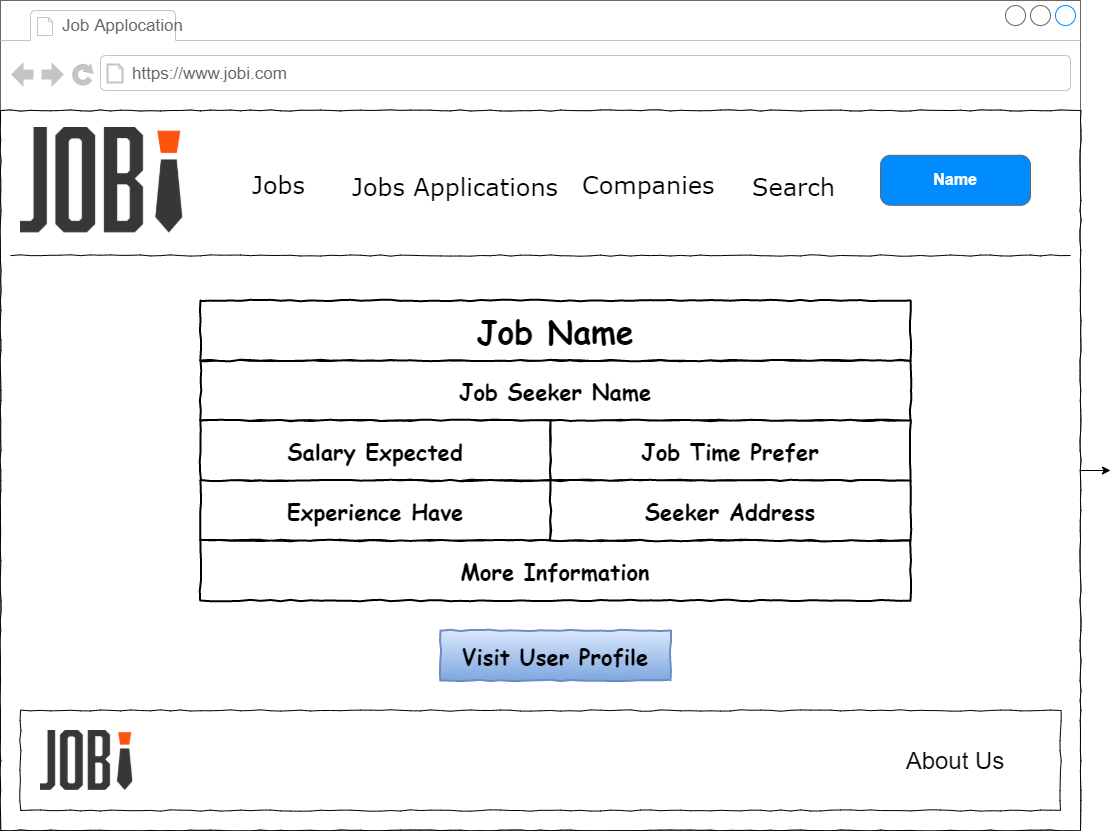
*Job Page*

**

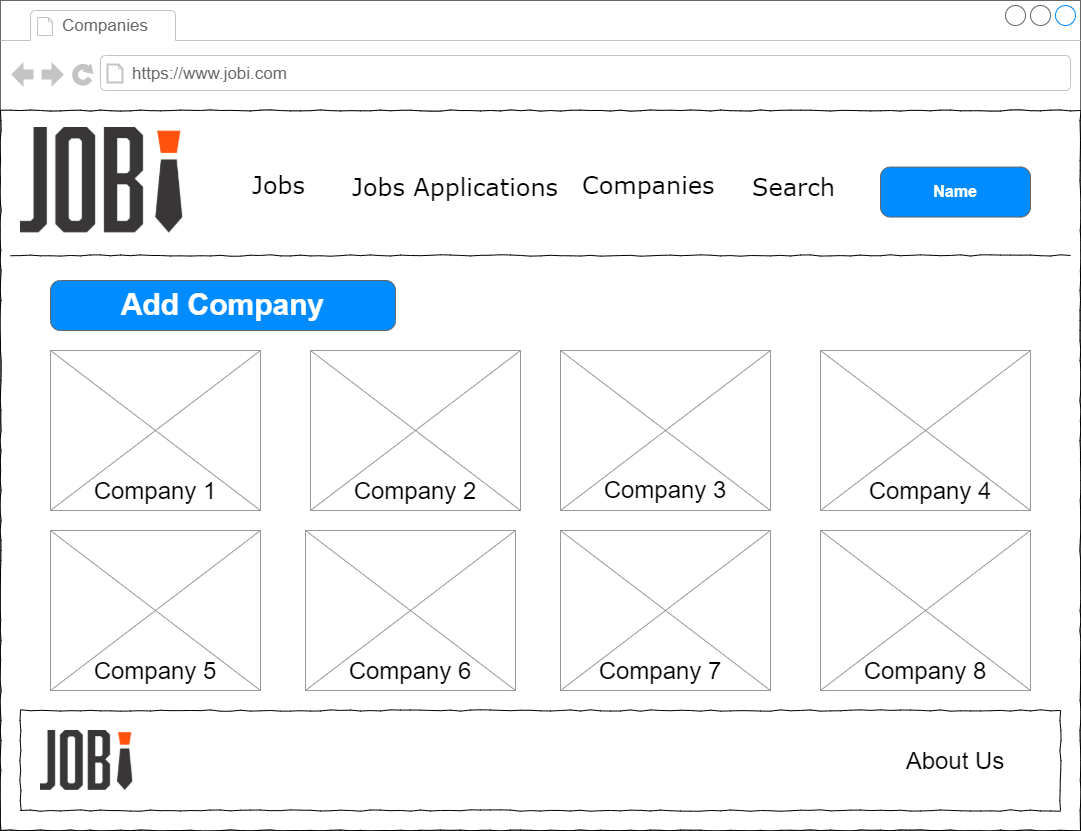
*Jobs Applications Page*



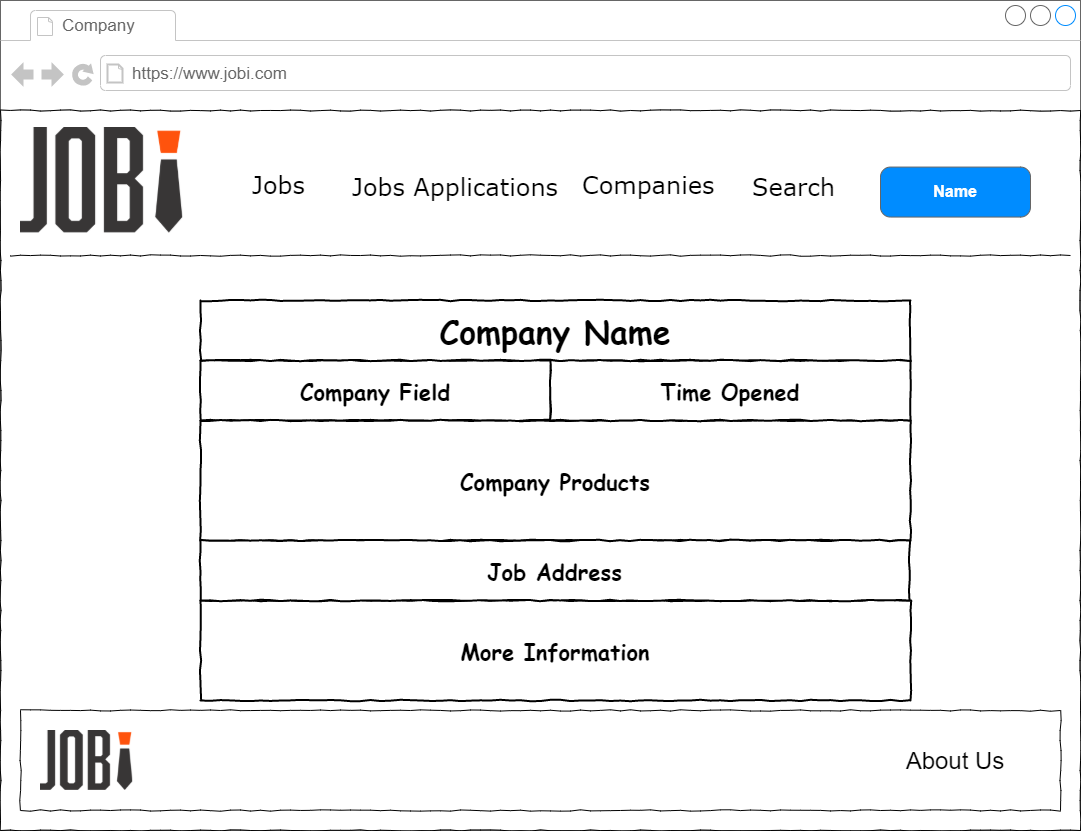
*Job Application Page*

**

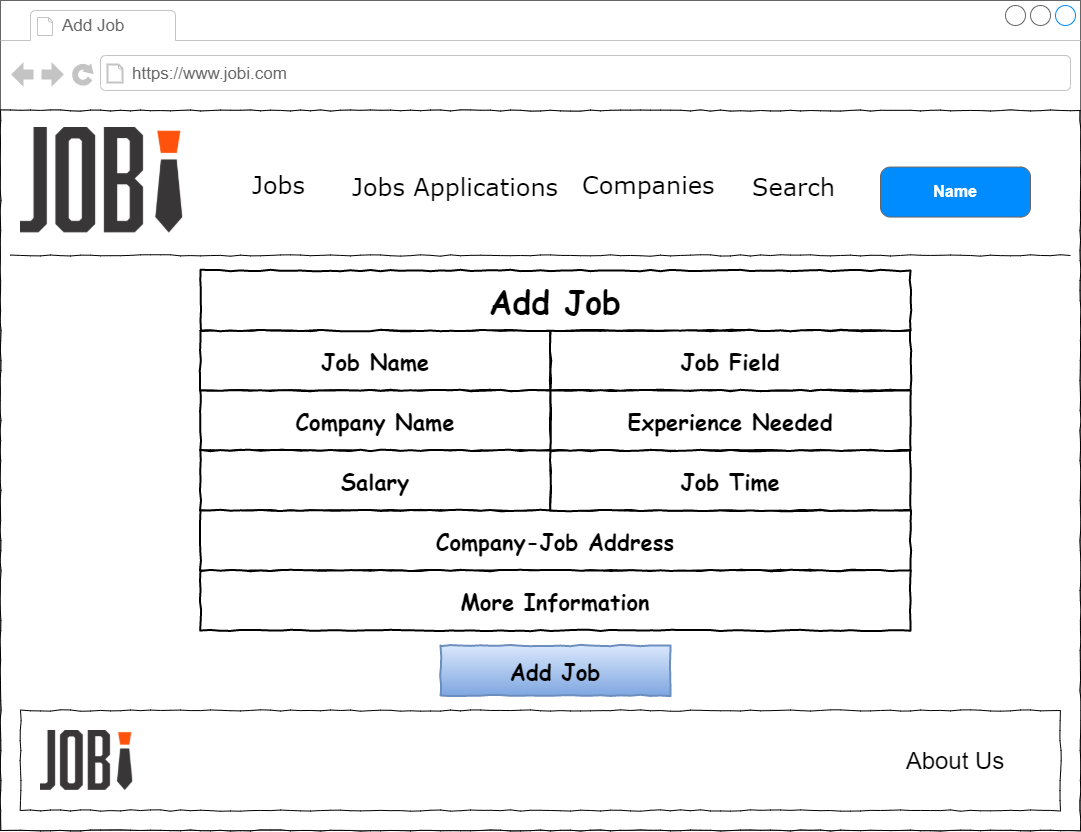
*Companies Page*

**

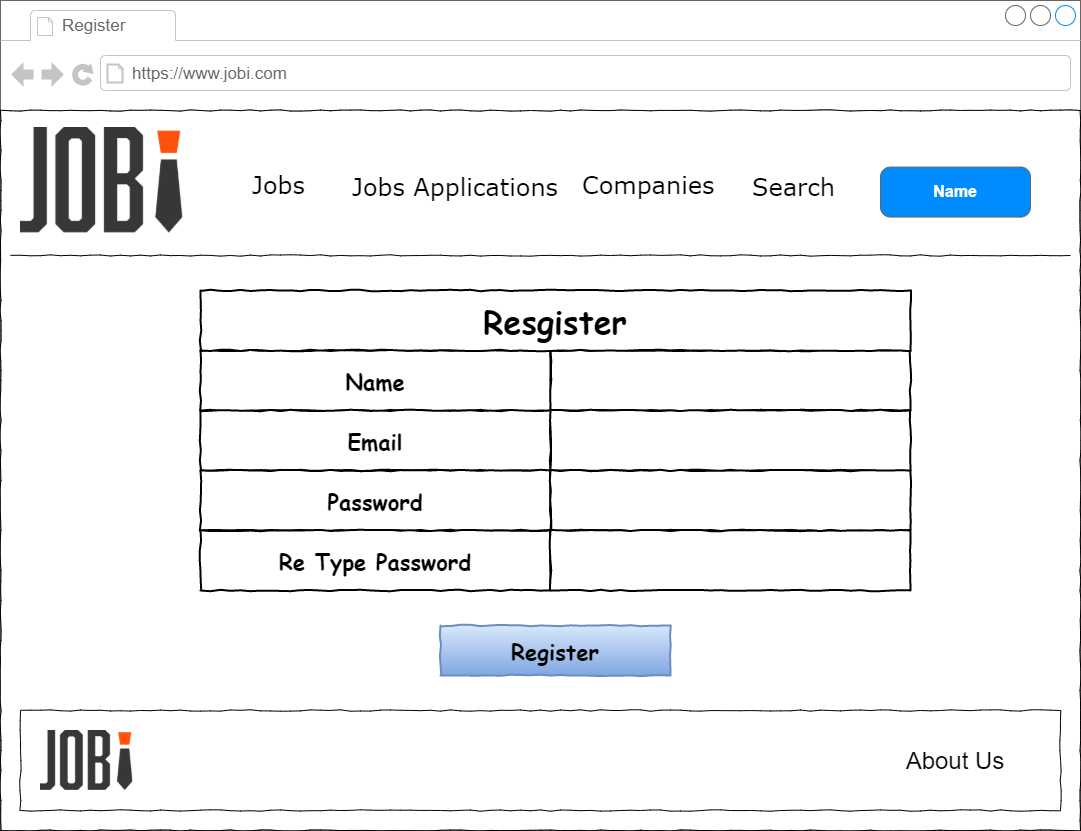
*Company Page*

**

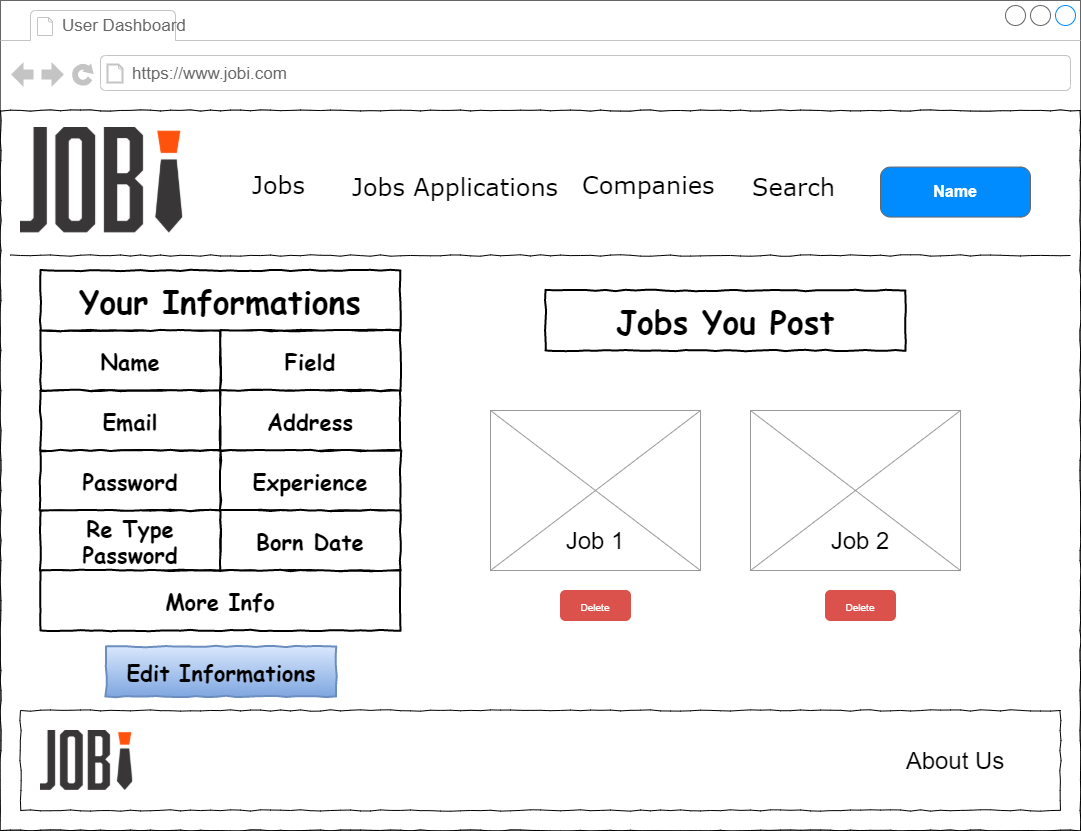
*Add Job Page*

**

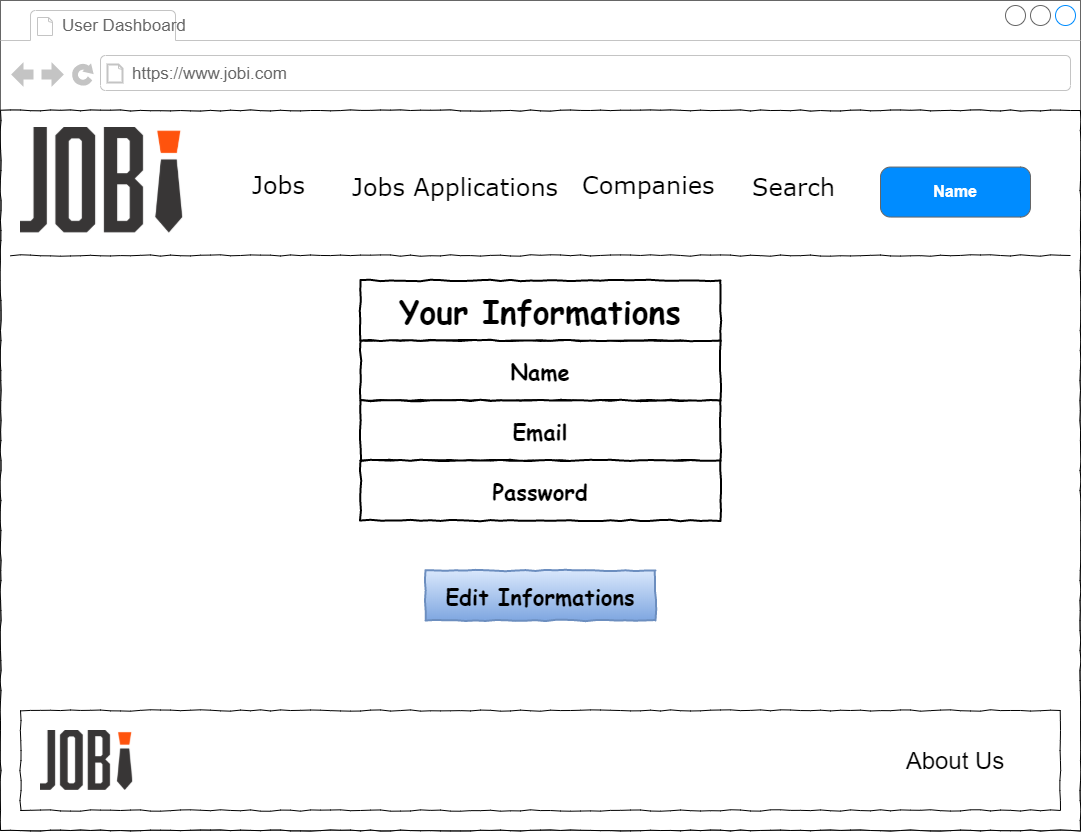
*Register Page*

**

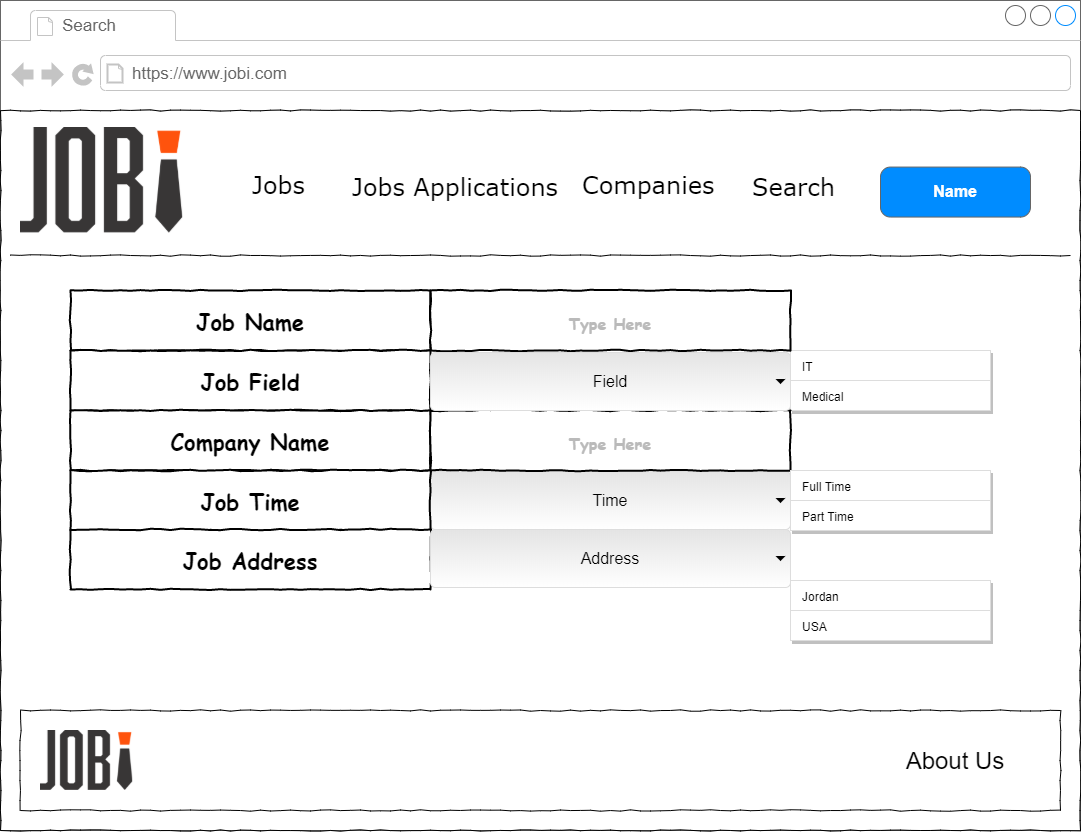
*User Dashboard Page*

**

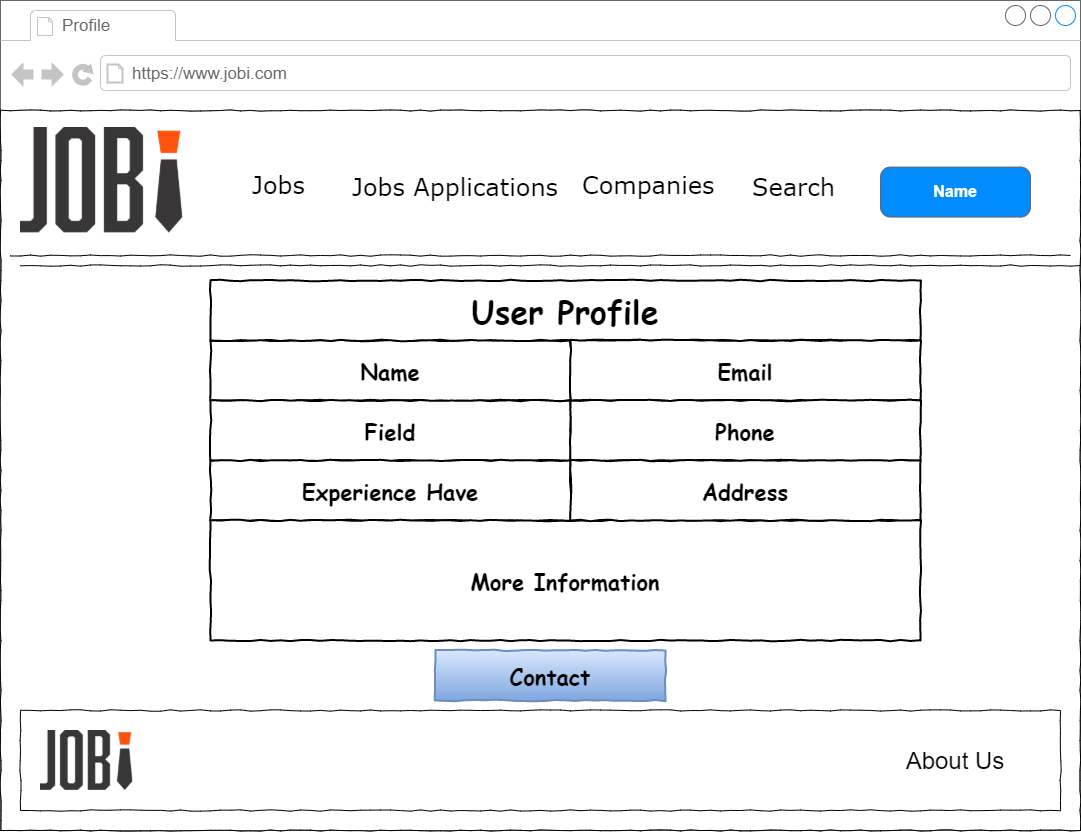
*Admin Dashboard Page*

**

*Search Page*

**

*Profile Page*

**

Chapter 4: Result & Recommendation

When project finish

Reference

1. <https://reactjs.org/docs/getting-started.html>
2. <https://stackoverflow.com/>
3. <https://github.com/>