**CSLAB Web-based Database**

**Architecture/Design Documentation**

**Table of Contents**

**1 INTRODUCTION…………………………………………………………………...2**

**2 DESIGN PROPERTIES…………………………………………………………….3**

**3 SYSTEM BEHAVIOR………………………………………………………………3**

**4 LOGICAL VIEW……………………………………………………………………4**

**4.1 High-Level Design (Architecture)…………………………………..4**

**5 CASE VIEW…………………………………………………………………………5**

**5.1 screenshot of the Interface …………………………………………6**

# Introduction

This document describes the architecture and design for the web-based database application being developed for Cybersecurity Research Lab. It is like an online archive for members of Cybersecurity Research Lab. New interns fill in their details and it is saved in the database.

The purpose of this document is to describe the architecture and design of the web-based database application for easy usage by the stakeholders and future updating. For this application the major stakeholders are:

* User – want assurances that the architecture will provide for system functionality and exhibit desirable non-functional quality requirements such as usability, reliability, etc.
* Developers – they want an architecture that will minimize complexity and development effort.
* Maintenance Programmers – they want assurance that the system will be easy to evolve and maintain on into the future.

The architecture and design for a software system is complex and individual stakeholders often have specialized interests. There is no one diagram or model that can easily express a system’s architecture and design. For this reason, software architecture and design is often presented in terms of multiple views or perspectives [IEEE Std. 1471]. Here the architecture of the web-based database application is described from 4 different perspectives [1995 Krutchen]:

1. Logical View – major components, their attributes and operations. This view also includes relationships between components and their interactions.
2. Case View – the case view is used to both motivate and validate design activity. It should be possible to walk through a case scenario and follow the interaction between high-level components. The components should have all the necessary behavior to conceptually execute a use case.

# Design Properties

The design priorities for the web-based database application are:

* The design minimized complexity and development effort.
* The design is easy to use.
* The design is adjustable.

# System Behavior

The application has two parts:

* The registration page where new interns register their contacts
* The administration page where the details of all the data are managed

# Logical View

In this section the modules of the system are expressed in terms of high level components (architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.

## High-Level Design (Architecture)

The website is written with html, css, javascript, php and mysql. The architecture consists of 2 major components:

* The front-end design(UI/UX)
* The back-end design(database)

Database

* The **Registration page** for accepting the details of new interns
* The **Database** is a central repository for data.
* The **Administrative page** where all the members are displayed

# Case View

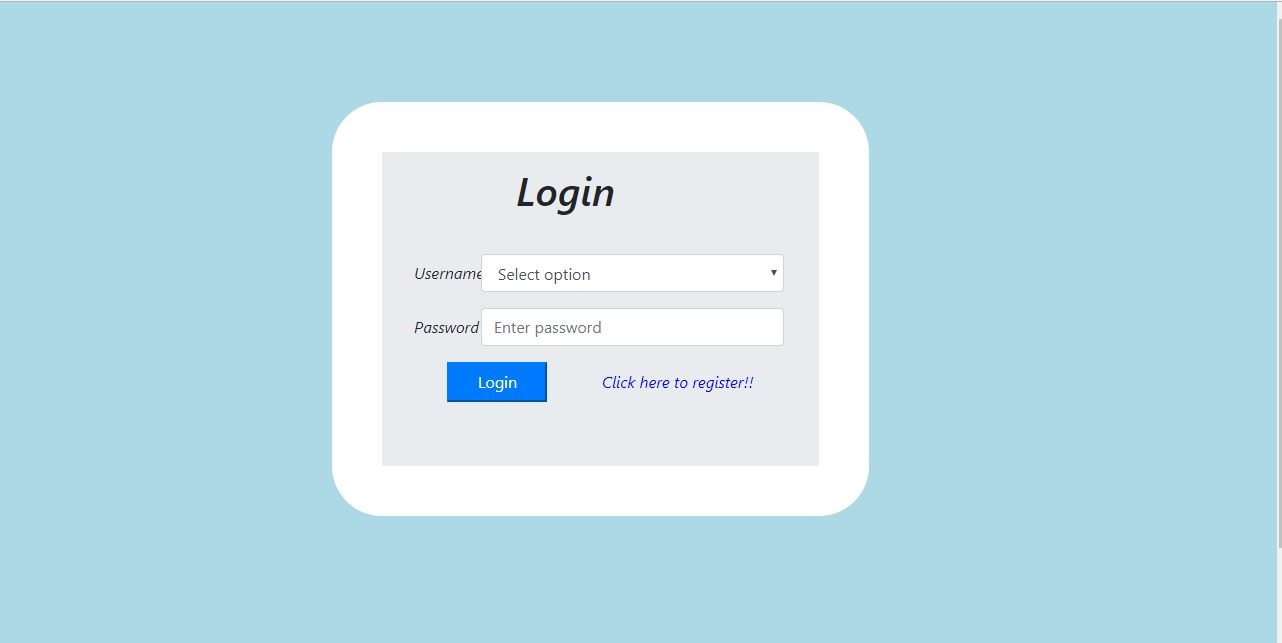
The Folder name for the project is called “**cslab\_doc**”. Which has subfolders and some php file that handles the login page

The subfolder names are

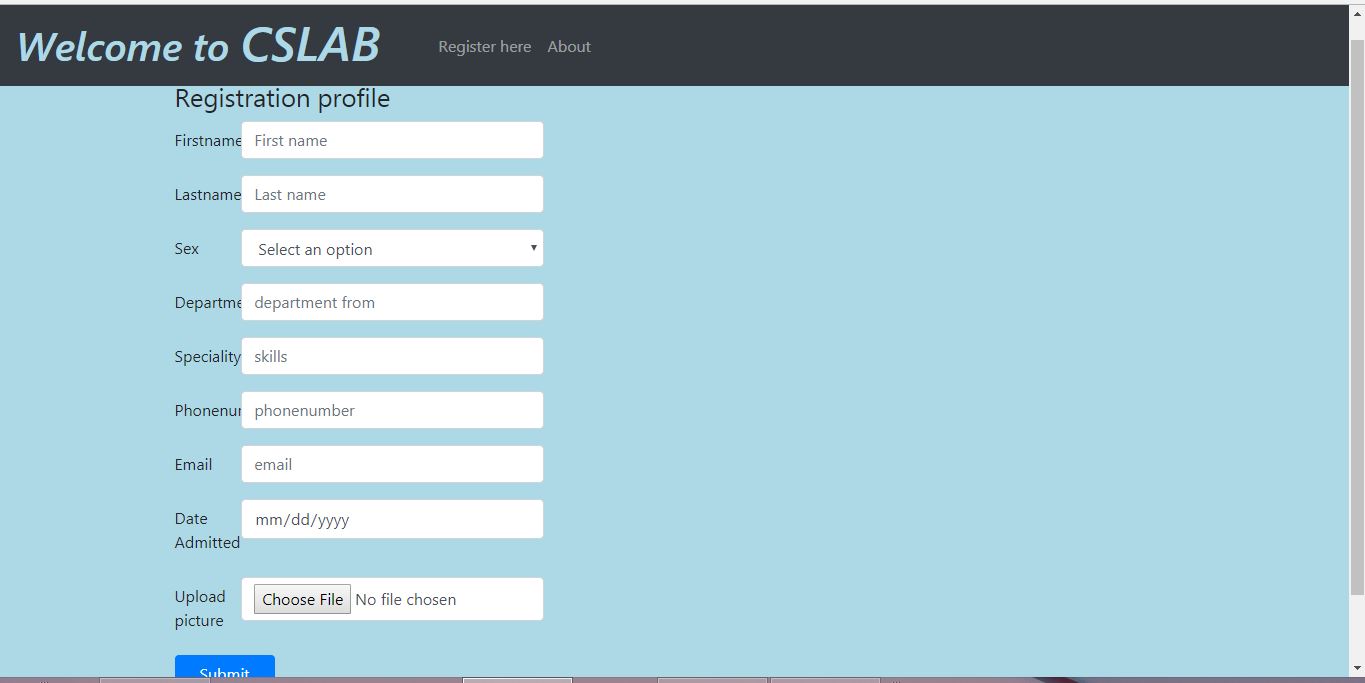
**Home**: Which contains code for the home page and a picture for the home page

**Profile\_mem**: Has a subfolder called “**Uploads**” (where images uploaded are stored) and php files that is designed to accept data from users, processes the data then stores it in the database

## 1.1 Screenshot of the interface



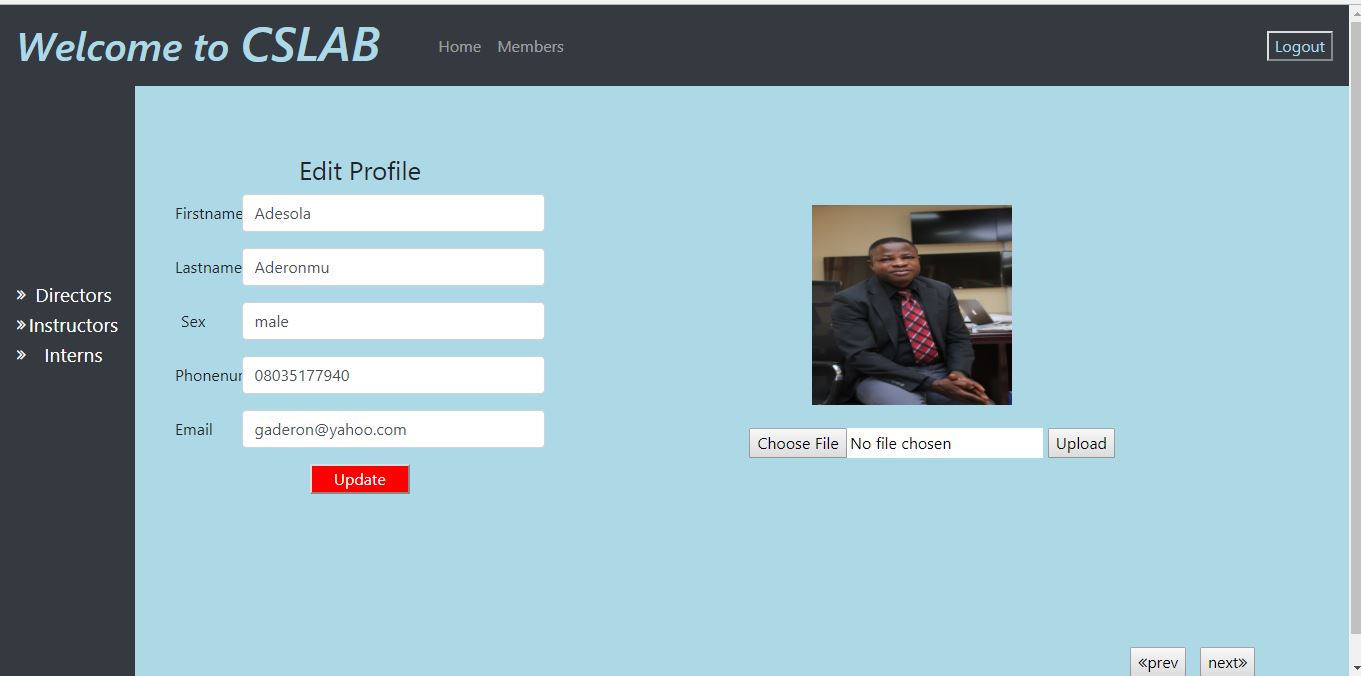
**Fig1**. **Login page**



**Fig2. Registration page**



**Fig3.** **Admin home page**



**Fig4. Profile page of each member**