

Resume Builder

A PROJECT REPORT

Submitted By

Madhur Maheshwari
(University Roll No: - 2000290140063)
Sakshi Panwar
(University Roll No: - 2000290140109)
Nistha Goyal
(University Roll No: - 2000290140077)

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATIONS

**Under the Supervision of
Mr. Prashant Agarwal
Associate Professor**



Submitted to

**DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(JAN 2022)**

CERTIFICATE

Certified that **Madhur Maheshwari <200029014005748>**, **Sakshi Panwar<200029014005794>**, **Nistha Goyal<200029014005762>** have carried out the project work having “**Resume Builder**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Madhur Maheshwari (2000290140063)

Sakshi Panwar (2000290140109)

Nistha Goyal (2000290140077)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Mr. Prashant Agarwal
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner

Signature of External Examiner

Dr. Ajay Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Mr. Prashant Agarwal** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Madhur Maheshwari

Sakshi Panwar

Nistha Goyal

List of Content

1 Introduction.....	06
1.1 Purpose.....	06
1.2 Scope	06
1.3 Overview.....	07
1.4 Goals Of proposed System.....	07
1.5 Background.....	08
1.6 Project Requirements.....	09
1.7 Technology Used.....	09
2 Feasible Study.....	10-12
2.1 Technical Feasibility.....	10
2.2 Economical Feasibility.....	11
2.3 Operational Feasibility.....	12
2.4 Schedule Feasibility.....	12
3 Architectural Design.....	13-18
3.1 ER diagram.....	13-14
3.2 Data flow diagram.....	15-16
3.3 Use Case Diagram.....	17-18
4 System Design.....	19-21
4.1 User table.....	19
4.2 Personal Table.....	20
4.3 Education Table.....	20
4.4 Other Table.....	21
5 Implementation.....	22-44
5.1 Code.....	22-39
5.1.1 Sign up.....	22-24
5.1.2 Login.....	24-26
5.1.3 Personal.....	26-29
5.1.4 Education.....	29-33
5.1.5 Experience.....	33-37
5.1.6 Project.....	37-39
5.2 Test case.....	40
5.3 Snapshot.....	41-43
5.3.1 Sign up.....	41
5.3.2 Login.....	41
5.3.3 Personal.....	42
5.3.4 Education.....	42

5.3.5	Experience.....	43
5.3.6	Project.....	43

LIST OF TABLES

Table No.	Name of Table	Page
4.1	User Table	
4.2	Personal Table	
4.3	Educational Table	
4.4	Other Table	

Introduction

Resume is the first meeting between you and a prospective employer more often now than ever. So, how do you want to be remembered? Wrinkled and unorganized. Neat and structured. Long and boring. Precise and interesting. Companies do not have the time to interview every applicant that is interested in the job. If they did, there would not be a company to work for. They use an eliminating process. That's right - resumes. When a job seeker wants to apply for a job online then generally, he/she needs to attach his/her resume with the email. Online Resume Building System provides the users the popular resume formats & a better way to show their resumes to the employers. A job seeker does not need to attach a resume with every email, he/she just must include the URL of his/her resume and the employer can view the resume online by clicking on the link and can download as well.

1.1 Purpose

Purpose of Online Resume Builder is to provide a way to the customers to design their resumes according to their requirements.

- a) Creating resumes online.
- b) Customizing the look and details.

1.2 Scope

Online Resume Builder can be used in accordance with the requirements of the customers. Customers can customize their resumes with their choice of themes & details. The services are hard to be defeated by the competitors as the system is providing the customers exactly what they want.

1.3 Overview

Project is related to Online Resume Building.

- This project maintains 1 types of users.
- Users (Customers)

Facilities provided by this project are as follows

- Details of customers are recorded.
- Update of data is easy.
- Flow of information is fast and easy.
- Customers can login to their accounts and view & update their data.

1.4 Goals of Proposed System

1) Planned approach towards working: - The working of the system will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

2) Accuracy: - The level of accuracy in the proposed system will be higher. All operation would be done correctly, and it ensures that whatever information is coming from the system is accurate.

3) Reliability: - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.

4) No redundancy: - In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.

5) Immediate Retrieval of Information: - The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.

6) Immediate storage of information: - In manual system there are many problems to store & update the large amount of information

7) Easy to operate: - The system should be easy to operate and should be such that it can be easily understood by a new user.

1.5 Background

Online Resume Builder is a system which allows the customers to design their resumes in accordance with their requirements. System provides facilities like...

- Customizing the resumes according to the user requirements.
- Editing the design.
- Choosing from latest professional designers.
- Viewed resume notification for the customers.

Various operations done in the system are as follows...

- Registering customers.
- Access to viewers/employers is allowed for public/shared data only. Writing resumes.
- Editing in design.
- Keeping track of latest formats of resumes. Viewed resume notification

1.6 Project Requirements

Software requirements	
Operating system	Software requirement
Windows 7,8,10, Linux, or any other higher version	Google chrome, internet explorer, or any web browser

Hardware requirements	
Processor	RAM
Core i3,i5 or i7	2GB or more

1.7 Technologies used:

This project will be an Internet application to be developed in following tools and technologies.

- Front End
 - Web technology: - React.js, JavaScript
 - Languages used: - JavaScript, HTML5, CSS
 - Development tool: - Sublime text
- Back End
 - Node.js, MongoDB

2.0 Feasibility Study

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. “FEASIBILITY STUDY” is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions: ☐ What are the user’s demonstrable needs and how does a system meet them? ☐ What resources are available for given system? ☐ What are the likely impacts of the system on the organization? ☐ Whether it is worth to solve the problem? During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this. Steps in feasibility analysis

Eight steps involved in the feasibility analysis are: ☐ Form a project team and appoint a project leader

Prepare system flowcharts.

Enumerate potential proposed system.

Define and identify characteristics of proposed system.

Determine and evaluate performance and cost effectiveness of each proposed system.

Weight system performance and cost data.

Select the best-proposed system. Prepare and report final project directive to management.

2.1 Technical Feasibility

Technical feasibility is the study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. Can the work for the project be done with current equipment existing software technology & available personal?

- Can the system be upgraded if developed?
- If new technology is needed, then what can be developed? This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may include Front-end and back-end selection An important issue for the development of a project is the selection of suitable

front-end and back-end. When we decided to develop the project, we went through an extensive study to determine the most suitable platform that suits the needs of the academy as well as helps in development of the project. The aspects of our study included the following factors. Front-end selection:

- It must have a graphical user interface that assists users that are not an advanced user of computer.
- Scalability and extensibility.
- Flexibility.
- Robustness.
- According to the organization requirement and the culture
- Must provide excellent reporting features with good printing support.
- Platform independent.
- Easy to debug and maintain.
- Event driven programming facility.
- Front end must support some popular back end like Ms Access

2.2 Economical Feasibility

Economic justification is generally the “Bottom Line” consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization i.e., profit making, the project is making to, the analysis and design phase

The financial and the economic questions during the preliminary investigation are verified to estimate the following:

- ☐ The cost to conduct a full system investigation.
- ☐ The cost of hardware and software for the class of application being considered.
- ☐ The benefits in the form of reduced cost.
- ☐ The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.
- ☐ This feasibility checks whether the system can be developed with the available funds. Online Resume Builder does not require enormous amount of money to be developed. This can be done economically if planned judiciously, so it is economically feasible. The cost of project depends upon the number of man-hours required.

2.3 Operational Feasibility

It is mainly related to human organizations and political aspects. The points to be considered are:

What changes will be brought with the system?

What organization structures are disturbed?

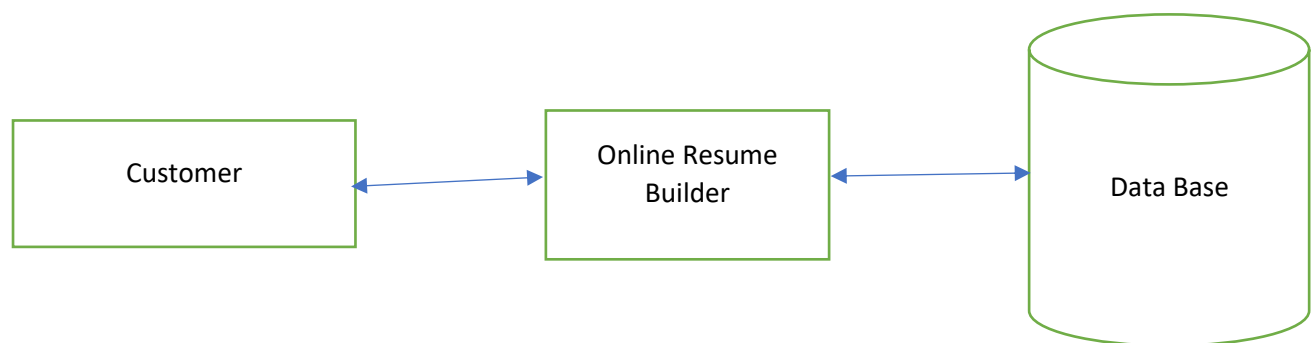
What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time? The system is operationally feasible as it very easy for the End users to operate it

2.4 Schedule Feasibility

Time evaluation is the most important consideration in the development of project. The time schedule required for the development of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems. Online Resume Builder can be developed in the considerable amount of time.

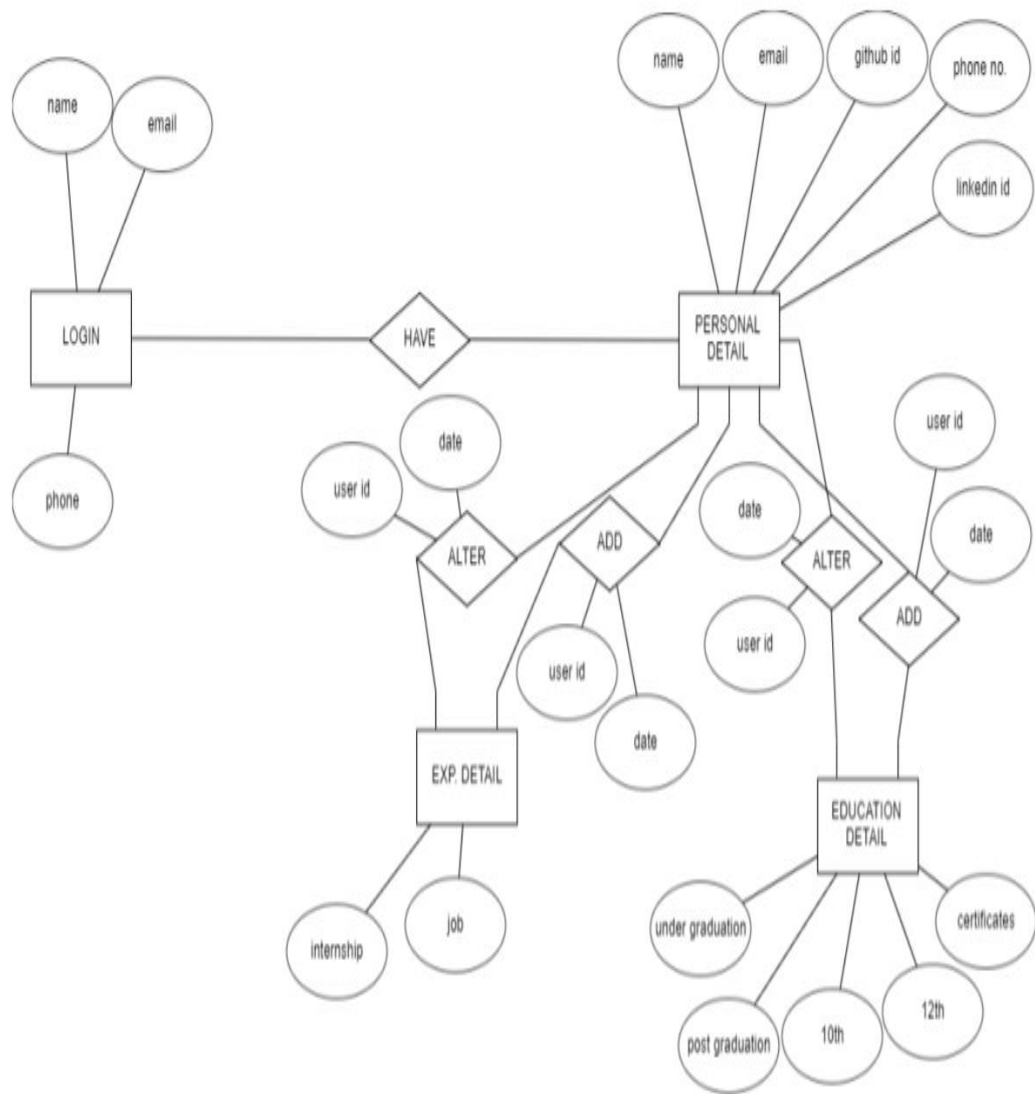
3. Architectural Design

Architectural design represents the structure of data and program components that are required to build a computer-based system. It considers the architectural style that the system will take, the structure and properties of the components that constitute the system, and the interrelationships that occur among all architectural components of a system.



ER Diagram

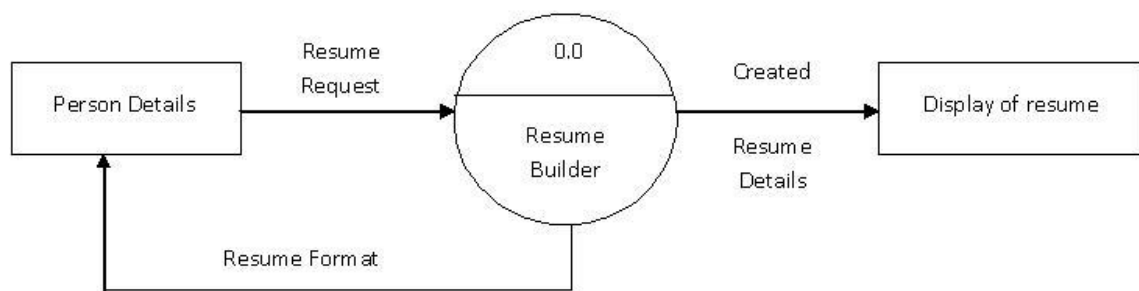
The object/relationship pair is the cornerstone of the data model. These pairs are represented graphically using E-R diagrams. A set of primary components are identified for the ERD: data objects, attributes, relationships, and various type indicators. The primary purpose of ERD is to represent data objects and their relationships



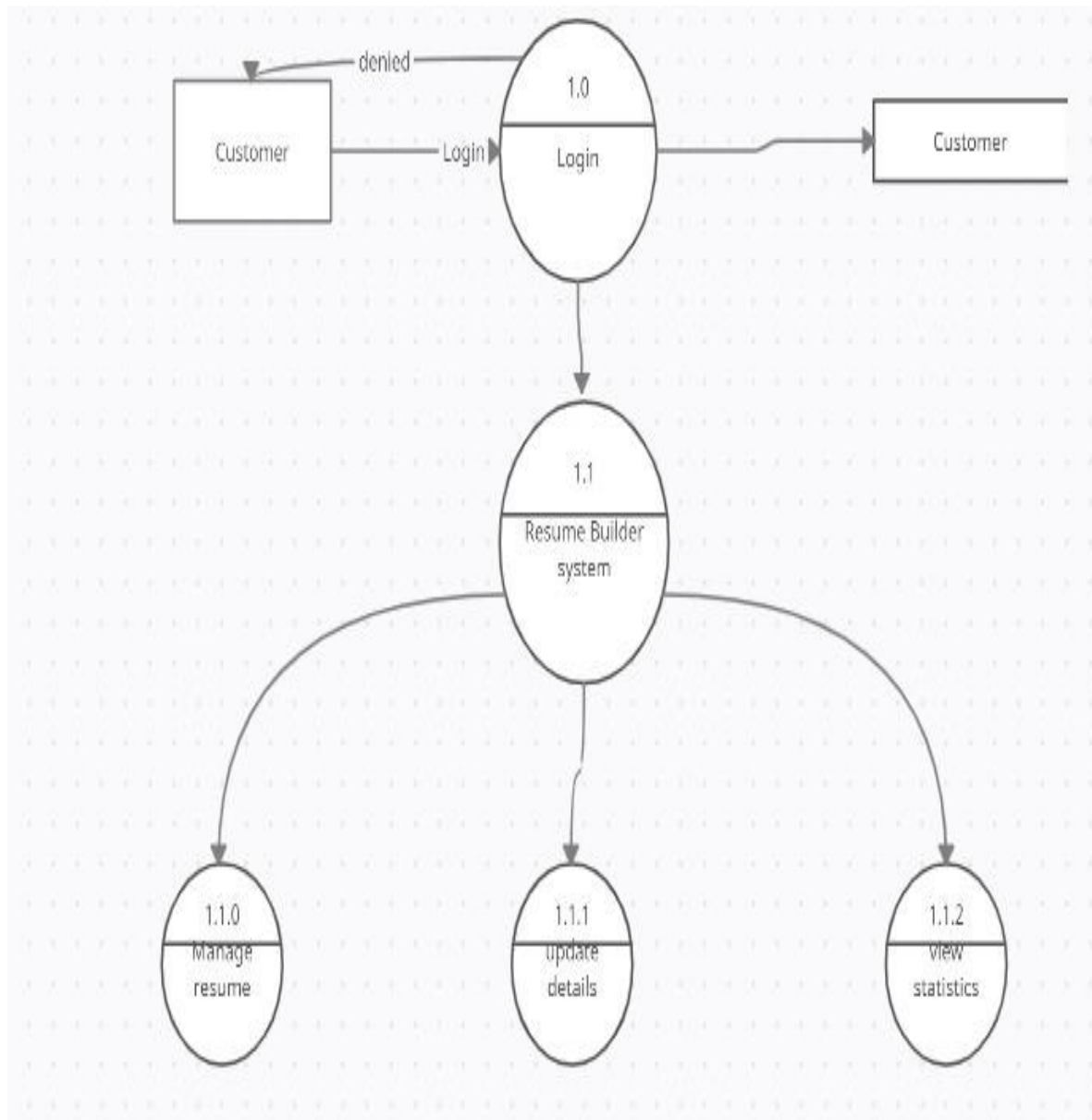
DFD (Data Flow Diagram)

The data flow diagram enables the software engineer to develop models of the information domain and functional domain at the same time. As the DFD is refined into greater level of detail, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in corresponding refinement of data as it moves through the processes that embody the application

0 level DFD:



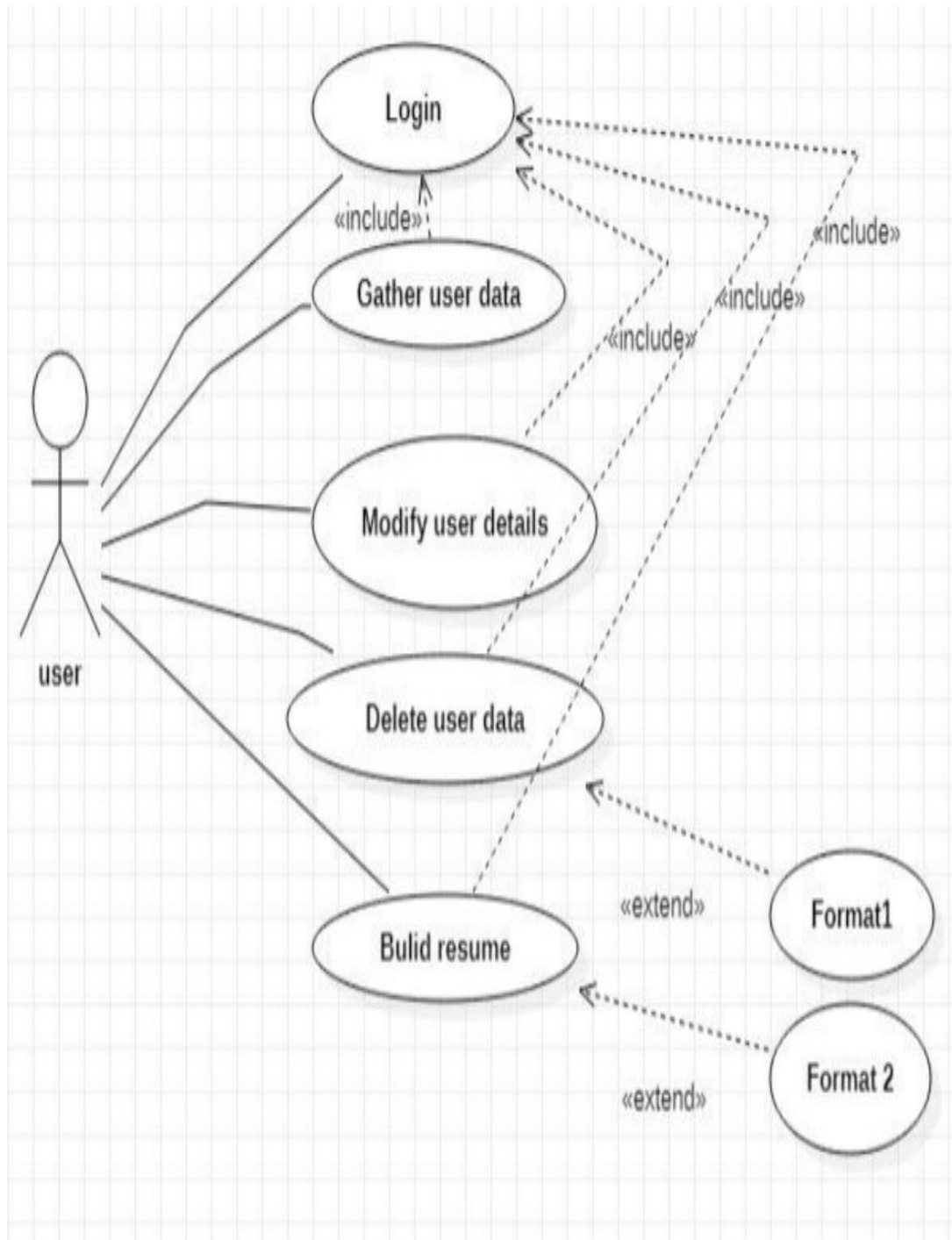
1 level DFD:



Use case Diagram:

Use Case Model is an approach that is a combination of text and pictures to improve the understanding of requirements. A use case model' is describing the complete functionality of a system by identifying how everything that is outside the system interacts with it.

A Use Case Diagram is given below that relates to this application.



4.0 System Design

System design is a solution, a 'how to' approach to the creation of a new system. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Emphasis is on translating the performance requirements into design specifications. Design goes through logical and physical stages of development. Logical design reviews the present physical system; prepared input and output specifications; details the implementation plan; and prepares a logical design walkthrough. The physical design maps out the details of the physical system, plans the system implementation, devises a test and implementation plan, and specifies any new hardware and software.

4.1 Database Tables

Users Table

Field name	Data type
Name	String
Email	String
Password	String

Personal Table

Field name	Data type
Name	String
Email	String
Linkedin	String
Github	String
Skill	String
Mobile	String

Education Table

Field name	Data type
School details	Array
College details	Array

Other Details

Field name	Data type
Project details	Array
Achievements	Array
Hobby	Array
Language	Array

5 Implementation

5.1 Code: -

Sign up

```
import React, { Component } from 'react'
import axios from 'axios'
class Signup extends Component {

  state:{
    name:''
  }

  login=e=>{
    e.preventDefault()
    window.location.href="/login"
  }

  signup = e => {
    e.preventDefault();
    console.log("signup.....",this.state)
    var {email,password,name}=this.state
    axios.post('/signup',{email,password,name}).then(resp=>{
      if(resp.data.success == 1)
      {
        alert(resp.data.message)
        window.location.href="/login"
      }
      else
      {
        alert("unable to save user")
        console.log(resp.data)
      }
    })
  }

  handleChange = ({ target: { value, name } }) => this.setState({
    [name]: value })
}
```

```

render() {

    return (
      <div className="card animated fadeInLeft col-lg-9
container">
        <div className="card-body">

          <h3 className="card-title text-
center">Signup</h3>
          <hr />
        </div>
        <form className="col-lg-9 container">
          <div className="form-group">
            <label for="exampleInputname">Name</label>
            <input type="text" className="form-control"
id="exampleInputname" onChange={this.handleChange} aria-
describedby="emailHelp" placeholder="Enter name"
name="name"/>

          </div>
          <div className="form-group">
            <label for="exampleInputEmail1">Email address</label>
            <input type="email" className="form-control"
id="exampleInputEmail1" onChange={this.handleChange} aria-
describedby="emailHelp" placeholder="Enter email"
name="email" />
            <small id="emailHelp" className="form-text text-
muted">We'll never share your email with anyone
else.</small>
          </div>
          <div className="form-group">
            <label for="exampleInputPassword1">Password</label>
            <input type="password" className="form-control"
id="exampleInputPassword1" onChange={this.handleChange}
placeholder="Password" name="password" />
          </div>

          <button type="submit" className="btn btn-primary"
onClick={this.signup}>Signup</button>

```

```

        <button type="submit" className="btn btn-primary"
        onClick={this.login}>Login</button>

    </form>
  </div>
  )
}
}

export default Signup;

```

Login

```

import React, { Component } from 'react'
import axios from 'axios';
class Login extends Component {

  state:{
    name:""
  }
  signup=e=>{
    e.preventDefault()
    window.location.href="/signup"
  }

  login = e => {
    e.preventDefault();
    console.log("login.....",this.state.email)
    var {email,password} =this.state
    axios.post('/login',{email,password}).then(resp=>{
      if(resp.data.success == 1)
      {
        localStorage.setItem('auth_data',
JSON.stringify(resp.data.user));
        localStorage.setItem('token',
JSON.stringify(resp.data.token));
        window.location.href="/"
        // window.location.reload()

```



```

    }
    else
    {
        alert("User does not exists")
        console.log(resp.data)
    }
})
}
handleChange = ({ target: { value, name } }) => this.setState({
[name]: value })
render() {

    return (
        <div className="card animated fadeInLeft col-lg-9
container">
            <div className="card-body">

                <h3 className="card-title text-center">Login</h3>
                <hr />
            </div>
            <form className="col-lg-9 container">
                <div className="form-group">
                    <label htmlFor="exampleInputEmail1">Email
address</label>
                    <input type="email" className="form-control"
onChange={this.handleChange} name="email"
placeholder="Enter email"/>
                    <small className="form-text text-muted">We'll never
share your email with anyone else.</small>
                </div>
                <div className="form-group">
                    <label
htmlFor="exampleInputPassword1">Password</label>
                    <input type="password" className="form-control"
onChange={this.handleChange} placeholder="Password"
name="password" />
                </div>
            </form>
        </div>
    )
}

```

```

        <button type="submit" className="btn btn-primary"
onClick={this.login}>Login</button>
        <button type="submit" className="btn btn-primary"
onClick={this.signup}>Signup</button>

    </form>
  </div>
  )
}
}

export default Login;

```

Personal Details

```

import React, { Component } from 'react'
import axios from 'axios'
class PersonalDetails extends Component {
  state:{
    show_name:""
  }

  componentDidMount()
  {
    try
    {

      // var user=JSON.parse(localStorage.getItem("auth_data")).email
      var show_name=JSON.parse(localStorage.getItem("auth_data")).name
      console.log("show name is ",show_name)
      this.setState({show_name:show_name},()=>{
        console.log("show name updated")

      })

    }
    catch(e)
    {

```

```

        // user=""

    }
}

continue = e => {
    e.preventDefault();
    this.props.nextStep();

    console.log("this prosp.....",this.props)
    var {name,email,phone,github,linkedin,skills} = this.props.values
    try
    {

        var user=JSON.parse(localStorage.getItem("auth_data")).email
        var show_name=JSON.parse(localStorage.getItem("auth_data")).name
        console.log("show name is ",show_name)
        this.setState({show_name})

    }
    catch(e)
    {
        user=""

    }

    axios.post("/save/personal",{userid:user,name,email,mobile:phone,github,li
nkedin,skills}).then(data=>{
        console.log("data is ",data)
    }).catch(err=>{
        console.log("error is ",err)
    })
};

// savePersonal= ()=>{
//   console.log(this.props.values)
// }

```

```

render() {
  const { values, handleChange } = this.props;
  return (
    <div className="card animated fadeInLeft">
      <div className="card-body">

        <h3 className="card-title">Personal Info</h3>
        <hr />
      </div>
      <form onSubmit={this.continue}>
        <div className="row col-lg-10 mx-auto">
          <div className="col-lg-4 text-left">
            <label>Name*</label>
            <input type="text" name="name" className="form-
control" onChange={handleChange} defaultValue={values.status === 1 ? " :
values.name} required />
          </div>
          <div className="col-lg-4 text-left">
            <label>Email*</label>
            <input type="email" name="email" className="form-
control" onChange={handleChange} defaultValue={values.status === 1 ? " :
values.email} required />
          </div>
          <div className="col-lg-4 text-left">
            <label>Mobile*</label>
            <input type="text" name="phone" className="form-
control" onChange={handleChange} defaultValue={values.status === 1 ? " :
values.phone} required />
          </div>
        </div>
        <br />
        <div className="row col-lg-10 mx-auto">
          <div className="col-lg-6 text-left">
            <label>Linkedin</label>
            <input type="text" name="linkedin" className="form-
control" defaultValue={values.status === 1 ? " : values.linkedin}
onChange={handleChange} />
          </div>
          <div className="col-lg-6 text-left">

```

```

        <label>Github</label>
        <input type="text" name="github" className="form-
control" defaultValue={values.status === 1 ? " " : values.github}
onChange={handleChange} />
      </div>
    </div>
    <br />
    <div className="row col-lg-10 mx-auto">
      <div className="col-lg-12 text-left">
        <label>Skills* (Separate each skill with a space and a
comma)</label>
        <input type="text" name="skills" className="form-control"
defaultValue={values.status === 1 ? " " : values.skills}
onChange={handleChange} />
      </div>
    </div>
    <br />
    <div className="container text-center"><button type="submit"
className="btn btn-info" onClick={this.savePersonal}>Next<i
className="fas fa-angle-right ml-1"></i></button></div>
    <br />
  </form>
</div>
)
}
}

```

```
export default PersonalDetails;
```

Education

```

import React, { Component } from 'react'
import axios from 'axios'
export default class Education extends Component {

  back = e => {
    e.preventDefault();

```

```

    this.props.prevStep();
};

continue = e => {
  e.preventDefault();
  this.props.nextStep();
  try
  {

    var user=JSON.parse(localStorage.getItem("auth_data")).email
    console.log("user is -----0-----",user)
  }
  catch(e)
  {
    user=""

  }
  var
{exp1_org,exp1_pos,exp1_dur,exp1_desc,exp2_org,exp2_pos,exp2_dur,exp
2_desc,proj1_title,proj1_link,proj1_desc,proj2_title,proj2_link,proj2_desc,e
du1_year,edu1_school,edu1_qualification,edu1_desc,edu2_year,edu2_sch
ool,edu2_qualification,edu2_desc} = this.props.values

  axios.post("/save/education",{user,sch_yr_qual_desc:[`${edu1_year},${edu
1_school},${edu1_qualification},${edu1_desc}`,`${edu2_year},${edu2_scho
ol},${edu2_qualification},${edu2_desc}`],inst_pos_dur_desc:[`${exp1_org},${
exp1_pos},${exp1_dur},${exp1_desc}`,`${exp2_org},${exp2_pos},${exp2_d
ur},${exp2_desc}`],proj_title_url_desc:[`${proj1_title},${proj1_link},${proj1_
desc}`,`${proj2_title},${proj2_link},${proj2_desc}`]}).then(data=>{
  console.log("data is ",data)
}).catch(err=>{
  console.log("error is ",err)
})
}

render() {
  const { values, handleChange } = this.props;
  return (

```

```

<div className="card animated fadeInLeft">
  <div className="card-body">

    <h3 className="card-title">Education Info</h3>
    <hr />
  </div>
  <form onSubmit={this.continue}>

    <div className="row col-lg-10 mx-auto">
      <div className="col-lg-12 text-left">
        <h3><b><i className="fas fa-check-circle mr-
1"></i>1</b></h3>
      </div>
      <div className="col-lg-4 text-left">
        <label>College/University*</label>
        <input type="text" name="edu1_school" className="form-
control" defaultValue={values.status === 1 ? " " : values.edu1_school}
onChange={handleChange} required />
      </div>
      <div className="col-lg-4 text-left">
        <label>Year*</label>
        <input type="text" name="edu1_year" className="form-
control" defaultValue={values.status === 1 ? " " : values.edu1_year}
onChange={handleChange} required/>
      </div>
      <div className="col-lg-4 text-left">
        <label>Qualification*</label>
        <input type="text" name="edu1_qualification"
className="form-control" defaultValue={values.status === 1 ? " " :
values.edu1_qualification} onChange={handleChange} required/>
      </div>

    </div>
    <div className="row col-lg-10 mx-auto">
      <div className="col-lg-12 text-left">
        <label>Description*</label>

```

```

        <input type="text" name="edu1_desc" className="form-
control" defaultValue={values.status === 1 ? " : values.edu1_desc}
onChange={handleChange} required />

```

```

    </div>

```

```

</div>

```

```

<br />

```

```

    <div className="row col-lg-10 mx-auto">

```

```

        <div className="col-lg-12 text-left">

```

```

            <h3><b><i className="fas fa-check-circle mr-
1"></i>2</b></h3>

```

```

            <hr/>

```

```

        </div>

```

```

        <div className="col-lg-4 text-left">

```

```

            <label>School</label>

```

```

            <input type="text" name="edu2_school" className="form-
control" defaultValue={values.status === 1 ? " : values.edu2_school}
onChange={handleChange} />

```

```

        </div>

```

```

        <div className="col-lg-4 text-left">

```

```

            <label>Year</label>

```

```

            <input type="text" name="edu2_year" className="form-
control" defaultValue={values.status === 1 ? " : values.edu2_year}
onChange={handleChange} />

```

```

        </div>

```

```

        <div className="col-lg-4 text-left">

```

```

            <label>Qualification</label>

```

```

            <input type="text" name="edu2_qualification"
className="form-control" defaultValue={values.status === 1 ? " :
values.edu2_qualification} onChange={handleChange} />

```

```

        </div>

```

```

    </div>

```

```

    <div className="row col-lg-10 mx-auto">

```

```

        <div className="col-lg-12 text-left">

```



```

        <label>Description</label>
        <input type="text" name="edu2_desc" className="form-
control" defaultValue={values.status === 1 ? " " : values.edu2_desc}
onChange={handleChange} />
    </div>
</div>
<br />
<div className="container text-center">
    <button type="button" className="btn btn-info"
onClick={this.back}><i className="fas fa-angle-left mr-
1"></i>Back</button>
    <button type="submit" className="btn btn-info">Next<i
className="fas fa-angle-right ml-1"></i></button>
</div>
<br />

</form>

</div>
)
}
}

```

Experience

```
import React, { Component } from 'react';
```

```
import axios from 'axios'
```

```
class Experience extends Component {
```

```

    back = e => {
        e.preventDefault();
        this.props.prevStep();
    };

```

```

    continue = e => {
        e.preventDefault();
        this.props.nextStep();
        //      try

```

```

    // {

    // var user=JSON.parse(localStorage.getItem("auth_data")).email
    // console.log("user is -----0-----")
    =====,user)
    // }
    // catch(e)
    // {
    //   user=""

    // }
    // var
    {exp1_org,exp1_pos,exp1_dur,exp1_desc,exp2_org,exp2_pos,exp2_dur,exp
    2_desc,proj1_title,proj1_link,proj1_desc,proj2_title,proj2_link,proj2_desc}
    = this.props.values
    //
    axios.post("/save/education",{user,inst_pos_dur_desc:[`${exp1_org},${exp1
    _pos},${exp1_dur},${exp1_desc}`,`${exp2_org},${exp2_pos},${exp2_dur},${
    exp2_desc}`],proj_title_url_desc:[`${proj1_title},${proj1_link},${proj1_desc}
    `,`${proj2_title},${proj2_link},${proj2_desc}`]}).then(data=>{
      // console.log("data is ",data)
      // }).catch(err=>{
      //   console.log("error is ",err)
      // })

    }

```

```

render() {
  const { values, handleChange } = this.props;
  return (
    <div className="card animated fadeInLeft">
      <div className="card-body">

        <h3 className="card-title">Experience Info</h3>
        <hr />
      </div>
      <form onSubmit={this.continue}>

```

```

<div className="row col-lg-10 mx-auto">
  <div className="col-lg-12 text-left">
    <h3><b><i className="fas fa-check-circle mr-
1"></i>1</b></h3>
  </div>

  <div className="col-lg-4 text-left">
    <label>Institute/Organisation</label>
    <input type="text" name="exp1_org" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp1_org}
onChange={handleChange} />
  </div>
  <div className="col-lg-4 text-left">
    <label>Position</label>
    <input type="text" name="exp1_pos" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp1_pos}
onChange={handleChange} />
  </div>
  <div className="col-lg-4 text-left">
    <label>Duration</label>
    <input type="text" name="exp1_dur" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp1_dur}
onChange={handleChange} />
  </div>
</div>
<div className="row col-lg-10 mx-auto">
  <div className="col-lg-12 text-left">
    <label>Description</label>
    <input type="text" name="exp1_desc" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp1_desc}
onChange={handleChange} />
  </div>
</div>

<br/>

```

```

<div className="row col-lg-10 mx-auto">
  <div className="col-lg-12 text-left">
    <h3><b><i className="fas fa-check-circle mr-
1"></i>2</b></h3>
    <hr/>
  </div>
  <div className="col-lg-4 text-left">
    <label>Institute/Organisation</label>
    <input type="text" name="exp2_org" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp2_org}
onChange={handleChange} />
  </div>
  <div className="col-lg-4 text-left">
    <label>Position</label>
    <input type="text" name="exp2_pos" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp2_pos}
onChange={handleChange} />
  </div>
  <div className="col-lg-4 text-left">
    <label>Duration</label>
    <input type="text" name="exp2_dur" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp2_dur}
onChange={handleChange} />
  </div>
</div>
<div className="row col-lg-10 mx-auto">
  <div className="col-lg-12 text-left">
    <label>Description</label>
    <input type="text" name="exp2_desc" className="form-
control" defaultValue={values.status === 1 ? " " : values.exp2_desc}
onChange={handleChange} />
  </div>
</div>
<br/>
<div className="container text-center">
  <button type="button" className="btn btn-info"
onClick={this.back}><i className="fas fa-angle-left mr-
1"></i>Back</button>

```

```

        <button type="submit" className="btn btn-info">Next<i
className="fas fa-angle-right ml-1"></i></button>
      </div>
    <br/>
  </form>
</div>
)
}
}

```

```
export default Experience;
```

Project

```

import React, { Component } from 'react'
import axios from 'axios'
class Project extends Component {

  back = e => {
    e.preventDefault();
    this.props.prevStep();
  };

  continue = e => {
    e.preventDefault();
    this.props.nextStep();
  }

  render() {
    const { values, handleChange } = this.props;
    return (
      <div className="card animated fadeInLeft">
        <div className="card-body">

          <h3 className="card-title">Projects' Info</h3>
          <hr />
        </div>

```

```
<form onSubmit={this.continue}>
```

```
    <div className="row col-lg-10 mx-auto">
      <div className="col-lg-12 text-left">
        <h3><b><i className="fas fa-check-circle mr-
1"></i>1</b></h3>
      </div>
      <div className="col-lg-6 text-left">
        <label>Title</label>
        <input type="text" name="proj1_title" className="form-
control" defaultValue={values.status === 1 ? " : values.proj1_title}
onChange={handleChange} />
      </div>
      <div className="col-lg-6 text-left">
        <label>Link</label>
        <input type="text" name="proj1_link" className="form-
control" defaultValue={values.status === 1 ? " : values.proj1_link}
onChange={handleChange} />
      </div>
```

```
    </div>
    <div className="row col-lg-10 mx-auto">
      <div className="col-lg-12 text-left">
        <label>Description</label>
        <input type="text" name="proj1_desc" className="form-
control" defaultValue={values.status === 1 ? " : values.proj1_desc}
onChange={handleChange} />
      </div>
    </div>
```

```
<br />
```

```
    <div className="row col-lg-10 mx-auto">
      <div className="col-lg-12 text-left">
        <h3><b><i className="fas fa-check-circle mr-
1"></i>2</b></h3>
```

```

        <hr/>
      </div>
      <div className="col-lg-6 text-left">
        <label>Title</label>
        <input type="text" name="proj2_title" className="form-
control" defaultValue={values.status === 1 ? " " : values.proj2_title}
onChange={handleChange} />
      </div>
      <div className="col-lg-6 text-left">
        <label>Link</label>
        <input type="text" name="proj2_link" className="form-
control" defaultValue={values.status === 1 ? " " : values.proj2_link}
onChange={handleChange} />
      </div>

    </div>
    <div className="row col-lg-10 mx-auto">
      <div className="col-lg-12 text-left">
        <label>Description</label>
        <input type="text" name="proj2_desc" className="form-
control" defaultValue={values.status === 1 ? " " : values.proj2_desc}
onChange={handleChange} />
      </div>
    </div>
    <br />
    <div className="container text-center">
      <button type="button" className="btn btn-info"
onClick={this.back}><i className="fas fa-angle-left mr-
1"></i>Back</button>
      <button type="submit" className="btn btn-info">Next<i
className="fas fa-angle-right ml-1"></i></button>
    </div>
    <br />
  </form>
</div>
)
}
}

```

```

export default Project;

```

5.2 Test case

Test Case Template									
Test Case Id	Component	Prior	Description/Test Suite	Pre-requisites	Test Steps	Expected Result	Actual Result	Status	Test Execution
Enter Invalid mail id and any password and then hit login button	Login_Page	P0	Verify that user has input a valid mail Id Example: 123@123.123, should be considered as invalid	Valid Email ID	1. Enter your Email ID 2. Enter your password 3. Press on Login Button	The email ID that you have entered doesn't match any account. Sign up for an	The email ID that you have entered doesn't match any account. Sign up for account		Team
Enter valid mail ID and incorrect password and then hit login button	Login_Page	P1	Verify that user has input a correct password that matches with an account in database	Correct Password	1. Enter your Email ID 2. Enter your password 3. Press on Login Button	The password you have entered is incorrect. forgotten password?	The password you have entered is entered incorrect. Forgotten	Pass	Team
Adding New Inventory	Add_Inventory_Page	P2	Check whether the quantity entered by user is not less than equals to zero	Qty>0	1. Enter Inventory Name 2. Enter Inventory Details 3. Enter Inventory Quantity 1. Enter your 10th details 2. Enter your 12th details 3. Enter your graduation details 4. Enter your post graduation details	Inventory quantity can't be less than or equal to zero	Inventory quantity can't be less than or equal to zero	Pass	Team
Enter all the education details	Eduaction details	P3	Verify that all the details will be filed	Eduaction details	1. Enter your name 2. Enter your phone no. 3. Enter your email Id	Inventory details should not be less than 1	Inventory details should not be less than 1	Pass	Team
Enter all the personal details	Personal details	P4	Verify that all the details will be filed	Personal details	1. Enter certificates 2. Enter internship	Inventory details should be matched	Inventory details should be matched	Pass	Team
Enter the Experience details	Experience details	P5	Verify that all the details will be filed	Experience details		Inventory details should be filled	Inventory details should be filled	Pass	Team

5.3 Snap Shot

Signup

Signup

Name

Email address

We'll never share your email with anyone else.

Password

Login

Login

Email address

We'll never share your email with anyone else.

Password

Personal details

Lets generate your Resume

Please provide accurate and clear description wherever necessary.

Personal Info

Name*

Email*

Mobile*

Linkedin

Github

Skills* (Separate each skill with a space and a comma)

NEXT >

Education

Education Info

✓ 1

College/University*

Year*

Qualification*

Description*

✓ 2

School

Year

Qualification

Description

< BACK

NEXT >

Experience

Experience Info

✓ 1

Institute/Organisation

Position

Duration

Description

✓ 2

Institute/Organisation

Position

Duration

Description

← BACK

NEXT →

Project info

Projects' Info

✓ 1

Title

Link

Description

✓ 2

Title

Link

Description

← BACK

NEXT →