

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
JNANA SANGAMA, BELAGAVI – 590 018



**A Mini Project Report  
on**

***ONLINE CV BUILDER***

*Submitted in partial fulfillment of the requirements as a part of the DBMS Lab for the V Semester of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi*

**Submitted by**

**SHREYA UDAY HASYAGAR  
1RN18IS101**

**SUPRIYA ALAVALA  
1RN18IS110**

**Guide**

**Mr. Ravikumar S G**  
Asst. Professor  
Dept. of ISE, RNSIT

**Lab In charge**

**Mr. R Rajkumar**  
Asst. Professor  
Dept. of ISE, RNSIT



**Department of Information Science and Engineering  
RNS Institute of Technology**  
Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,  
Bengaluru – 560 098  
2020 – 2021

# RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,  
Bengaluru – 560098

## DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING



## CERTIFICATE

This is to certify that the Mini project report entitled ***ONLINE CV BUILDER*** has been successfully completed by **SHREYA UDAY HASYAGAR** bearing USN **1RN18IS101** and **SUPRIYA ALAVALA** bearing USN **1RN18IS110**, presently V semester student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree **Bachelor of Engineering in Information Science and Engineering** under **Visvesvaraya Technological University, Belagavi** during academic year 2020 – 2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

---

**Mr. Ravikumar S G**  
GUIDE  
Assistant Professor

---

**Dr. S Sathish Kumar**  
Professor and HOD

### External Viva

#### Name of the Examiners

1. \_\_\_\_\_

2. \_\_\_\_\_

#### Signature with date

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## **ABSTRACT**

In the world full of opportunities, making a good impression is important in today's competitive job market and your CV will be your first contact with employers. A curriculum vitae is a marketing document that gives a summary of a job seekers career history, academic qualifications and also explains their future potential.

A completed CV is sent as an application to job advertisements or as a speculative approach to prospective companies. CVs are valuable and important because they are your first and maybe only direct communication with a potential employer. Presentation is the key. For this reason alone, it should be carefully thought out, designed and written.

People may find writing a CV to be boring, confusing and time-consuming. Online CV Builder is a unique system that takes input and stores the information in a database. This database fills up the information and you can create a document in CV format which shows you at your best abilities in few seconds and helps to get the job you want and deserve.

If a sheet of paper represents your entire work life, personality, and skills, it should better be a pretty amazing piece of paper.

## **ACKNOWLEDGMENT**

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

I would like to profoundly thank **Management of RNS Institute of Technology** for providing such a healthy environment to carry out this Mini Project work.

I would like to thank our beloved Director **Dr. H N Shivashankar** for his confidence feeling words and support for providing facilities throughout the course.

I would like to express my thanks to our Principal **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

I wish to place on record my words of gratitude to **Dr. S Sathish Kumar**, Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my Mini Project work.

I would like to express my profound and cordial gratitude to my guides, **Mr.Ravikumar S G** and **Mr. R Rajkumar**, Assistant Professor, Department of Information Science and Engineering for their valuable guidance in preparing Mini Project report.

I would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the Mini Project work.

And lastly, I would hereby acknowledge and thank my parents who have been a source of inspiration and also instrumental in carrying out this Mini Project work.

**SHREYA UDAY HASYAGAR SUPRIYA ALAVALA**

**USN: 1RN18IS101**

**USN: 1RN18IS110**

# TABLE OF CONTENTS

<b>CERTIFICATE</b>	ii
<b>ABSTRACT</b>	iii
<b>ACKNOWLEDGEMENT</b>	iv
<b>TABLE OF CONTENTS</b>	v
<b>LIST OF FIGURES</b>	vi
<b>ABBREVIATIONS</b>	vii
<b>1. INTRODUCTION</b>	1
1.1Background	1
1.2Introduction about the project	2
<b>2. E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM</b>	4
2.1Description of ER Diagram	4
2.1.1 Components of Online CV Builder, E-R Diagram	5
2.1.2 ER Diagram Relationships Description	5
2.2Description of Relational Schema Diagram	6
2.2.1 General Constraints	8
2.2.2 Schema Description	8
<b>3. SYSTEM DESIGN</b>	10
3.1Table Description	10
3.2Stored Procedure	14
3.3Triggers	15
<b>4. IMPLEMENTATION</b>	19
4.1Description of Tools and Technologies	19
4.2Front-end Development	19
4.2.1 Hypertext Markup Language	20
4.2.2 Cascading Style Sheets	20
4.2.3 JavaScript	21
4.2.4 Bootstrap	21
4.3Back-end Development	22
4.3.1 Backend Scripting language – PHP Hypertext Preprocessor	22
4.3.2 Web Server – Apache	23
4.3.3 Database - MySQL	23

4.4 User Flow Diagram	24
4.5 Discussion of Code Segment	24
4.6 Discussion of Results	64
4.7 Application of Project	73
<b>5. CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>74</b>
5.1 Conclusion	74
5.2 Future Enhancements	74
<b>REFERENCES</b>	<b>75</b>

# LIST OF FIGURES

<b>Figures</b>	<b>Descriptions</b>	<b>Page</b>
Figure. 1.1	Overview of DBMS	02
Figure. 2.1	E-R Diagram for Online CV Builder	04
Figure. 2.2	Relational Schema – Online CV Builder	07
Figure. 3.1	Stored Procedures	15
Figure. 3.2	Triggers	16
Figure. 4.1	Successful CV Build	24
Figure. 4.2	Homepage	64
Figure. 4.3	Homepage with Create CV link	64
Figure. 4.4	Sign up prompt	65
Figure. 4.5	Login prompt for existing users	65
Figure. 4.6	Login prompt for Admin	66
Figure. 4.7	Result of Trigger	66
Figure. 4.8	Options for the user	67
Figure. 4.9	Form to fill in necessary details	67
Figure. 4.10	Submission on filling mandatory fields	68
Figure. 4.11	Fields to choose for update	68
Figure. 4.12	Page if user chooses to update Education field	69
Figure. 4.13	Page if user chooses to update References field	69
Figure. 4.14	Options for Templates	70
Figure. 4.15	Output for Basic Template	70
Figure. 4.16	Output for Professional CV Template	71
Figure. 4.17	Delete CV	71

## **ABBREVIATIONS**

CV	-	Curriculum Vitae
DBMS	-	Database Management System
ER	-	Entity Relationship
DML	-	Data Manipulation Language
SQL	-	Structured Query Language
HTML	-	Hypertext Markup Language
CSS	-	Cascading Style Sheets
PHP	-	PHP Hypertext Processor
ECMA	-	European Computer manufacturers Association
JS	-	JavaScript
JSP	-	Java Server Pages
ASP	-	Active Server Pages
BSD	-	Berkeley Software Distribution

## CHAPTER-1

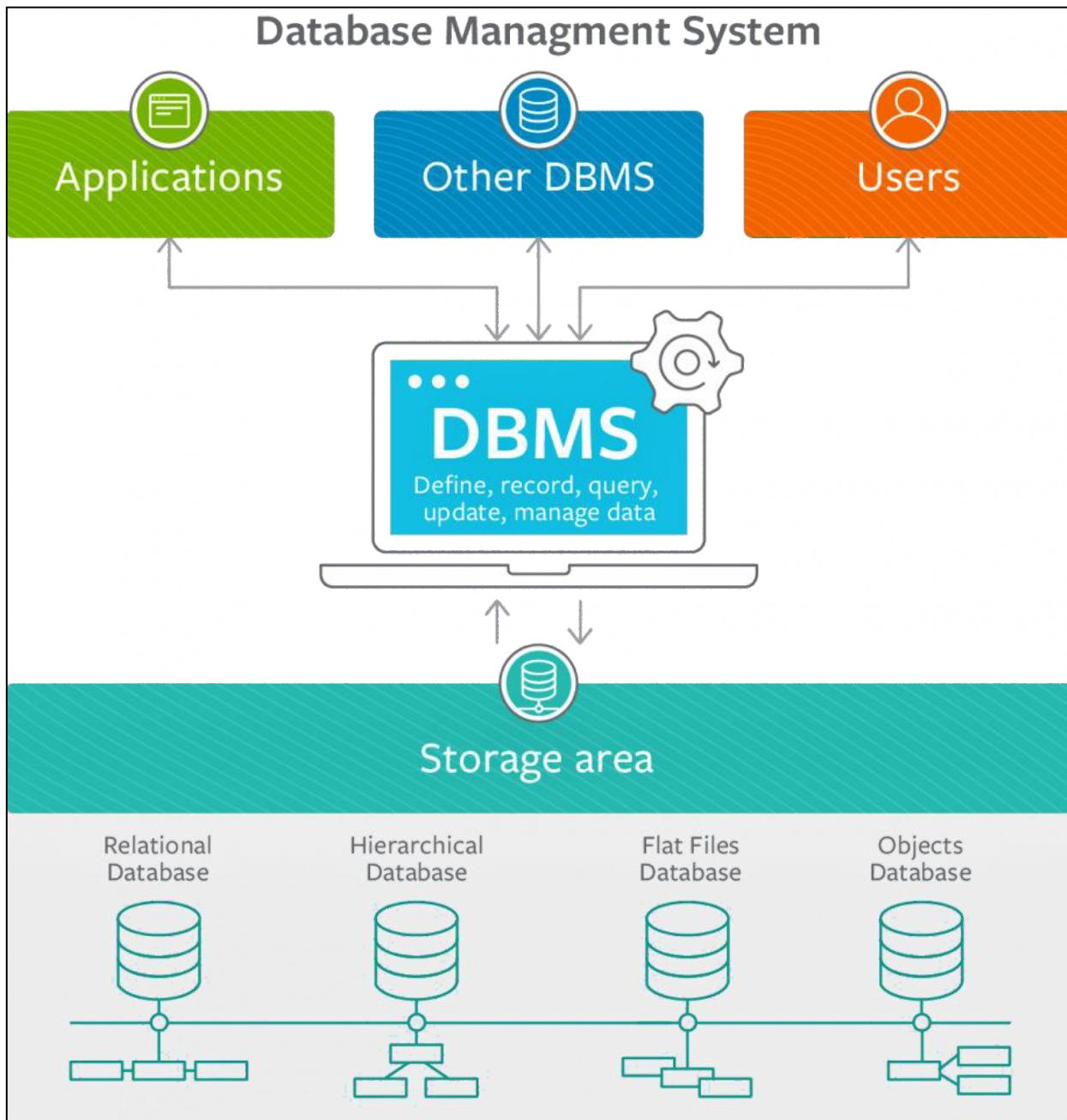
# INTRODUCTION

## 1.1 Background

**Data** is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. and can be stored in pieces of paper or electronic memory, etc.

A **database** is an organized collection of data, so that it can be easily accessed and managed. The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data. There are many databases available like MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, etc.

**Database Management System** (DBMS) is software for storing and retrieving users' data while considering appropriate security measures and consists of a group of programs which modify the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. DBMS software primarily functions as an interface between the end user and the database, simultaneously managing the data, the database engine, and the database schema in order to facilitate the organization and manipulation of data. Though functions of DBMS vary greatly, general-purpose DBMS features and capabilities should include: a user accessible catalog describing metadata, DBMS library management system, data abstraction and independence, data security, logging and auditing of activity, support for concurrency and transactions, support for authorization of access, access support from remote locations, DBMS data recovery support in the event of damage, and enforcement of constraints to ensure the data follows certain rules.



*Figure 1.1 Overview of DBMS*

## 1.2 Introduction to Online CV Builder

Online CV Builder is a service system which helps in building a new CV step-by-step for free. If the users are tired with formatting their application in a word processor or are pressed for time, then Online CV Builder helps cut down on how long it will take to make a CV on their own.

Online CV Builder helps in building an efficient CV for the user. The project takes essential information required to fill in the details of the user and stores it in the database. A CV begins with your contact information, including your name, address, telephone number, and email address. You should also indicate your area or areas of interest. Your CV should include a comprehensive account of your academic history. It must also contain details about all publications, research projects, and presentations to which you have contributed. You should also list any grants, academic awards, and other related honors you've received and the skills gained throughout your career. The employment and experience section of your CV should contain previous work positions, both paid and unpaid. In addition to jobs, include any relevant internships and volunteer experiences. Finally, provide a list of references, along with their contact information, on your curriculum vitae. The stored data in the database is used to construct a CV. The output of the same is provided in seconds.

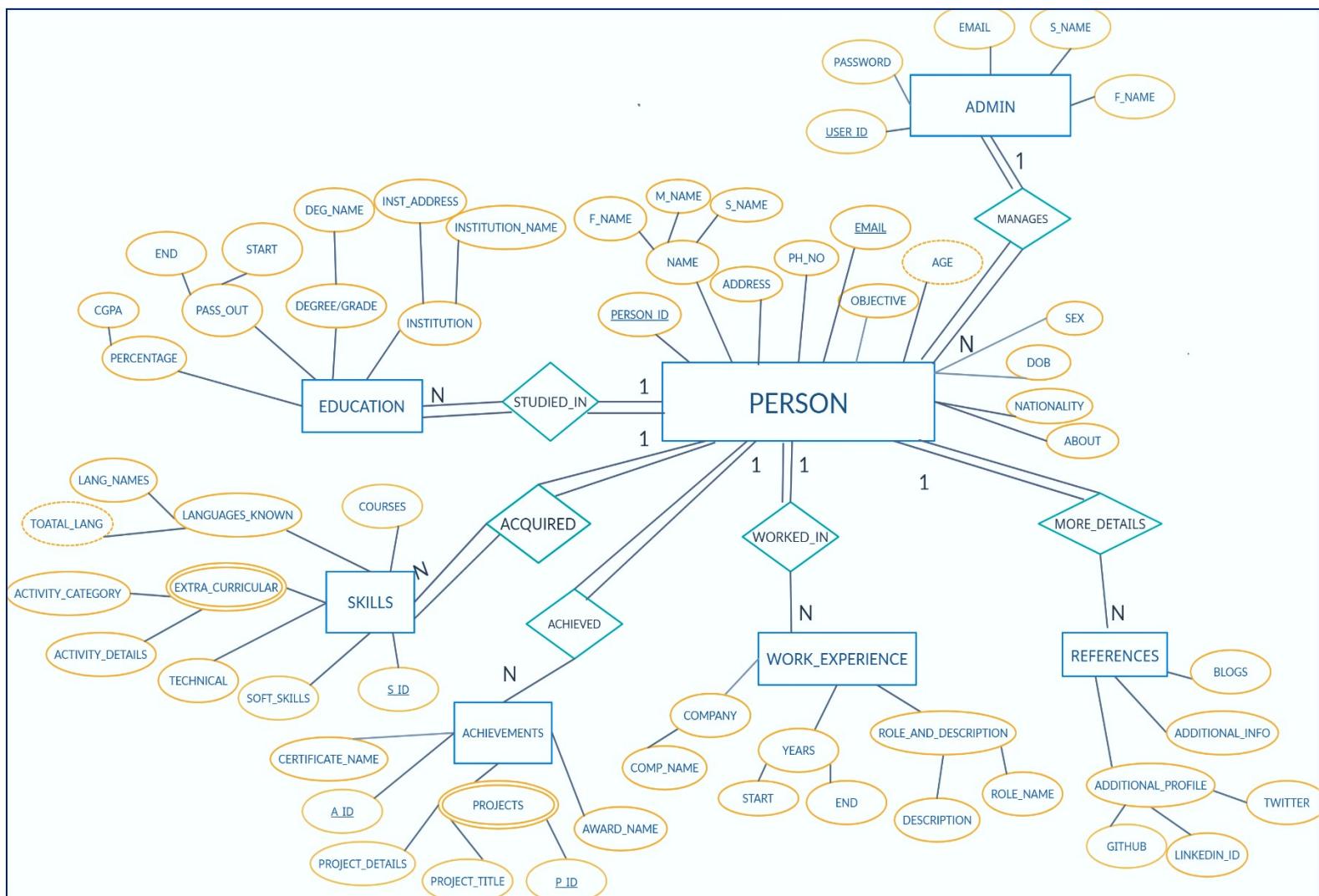
The user need not worry about the formatting or keeping the CV professional. The project helps the user to pick his choice between the templates and makes sure that the user includes the required fields compulsorily. It also provides the sample CV's to look into before inserting the information. The final CV can be downloaded in PDF format and used whenever and wherever needed.

## CHAPTER-2

# E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM

## 2.1 Description of ER Diagram

**ER Diagram** stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.



*Figure 2.1 ER Diagram*

## 2.1.1 Components of Online CV Builder, E-R Diagram

- An **entity** is an object or component of data. Entity types like **Admin, Person, Education, Work\_Experience, Skills, Achievements and References** are represented using rectangular boxes in ER Diagram.
- An **attribute** describes the property of an entity. Attributes like **Phone\_no, Email** etc. are represented as Oval attached to the entity type using a straight line in an ER diagram.
- **Key attributes** like **Person\_id** and **User\_id** is underlined within the oval.
- **Component attributes** of a composite attribute like **Name** are attached to oval representing it.
- **Multivalued attributes** like in **Projects** are represented by double layered oval.
- **Relationships** like **acquired, studied\_in** are represented in diamond boxes which are attached to the entity types participating in the relationship using straight lines.
- The **total participation** of the entities participating in the relationship represented inside the rhombus is identified by two straight lines from the entity type to the diamond. On the other hand, the **partial participation** is identified by a single line from the entity type to the diamond.

## 2.1.2 E-R Diagram Relationships Description

- 1) **ADMIN: PERSON** is of cardinality **1: N** as one admin can manage at least 1 or more user's account and therefore connected by **MANAGE** relationship. There is total participation from ADMIN as minimum of 1 admin is required to manage 1 or more PERSON.
- 2) **PERSON: EDUCATION** is of cardinality **1: N** as one person might have studied in 1 or more educational institutions and therefore connected by **STUDIED\_IN** relationship. There is total participation between the two entities as an individual might have acquired basic education from an institution.
- 3) **PERSON: SKILLS** is of cardinality **1: N** as one person might have gained 0 or more skills throughout his career and therefore connected by **ACQUIRED** relationship. There is total participation from PERSON there must be at least 1 person, but partial participation from SKILLS an individual might or might not have any skills.

- 4) **PERSON: ACHIEVEMENTS** is of cardinality **1: N** as one person might have achieved 0 or more things in life or worked on projects and therefore connected by **ACHIEVED** relationship. There is total participation from PERSON there must be at least 1 person, but partial participation from ACHIEVEMENTS as an individual might or might not have received any awards/certificates or worked on project.
- 5) **PERSON: WORK\_EXPERIENCE** is of cardinality **1: N** as one person might have worked in 0 or more companies and therefore connected by **WORKED\_IN** relationship. There is total participation from PERSON there must be at least 1 person, but partial participation from WORK\_EXPERIENCE as an individual might or might not have worked in any company
- 6) **PERSON: REFERENCES** is of cardinality **1: N** as one person might have worked in 0 or more references to provide and therefore connected by **MORE\_DETAILS** relationship. There is total participation from PERSON there must be at least 1 person, but partial participation from REFERENCES as an individual might or might not have more information to provide.

## 2.2 Description of Relational Schema Diagram

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A schema diagram is a diagram which contains entities and the attributes that will define that schema.

A database schema can be divided broadly into two categories –

- **Physical Database Schema** – this schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- **Logical Database Schema** – this schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

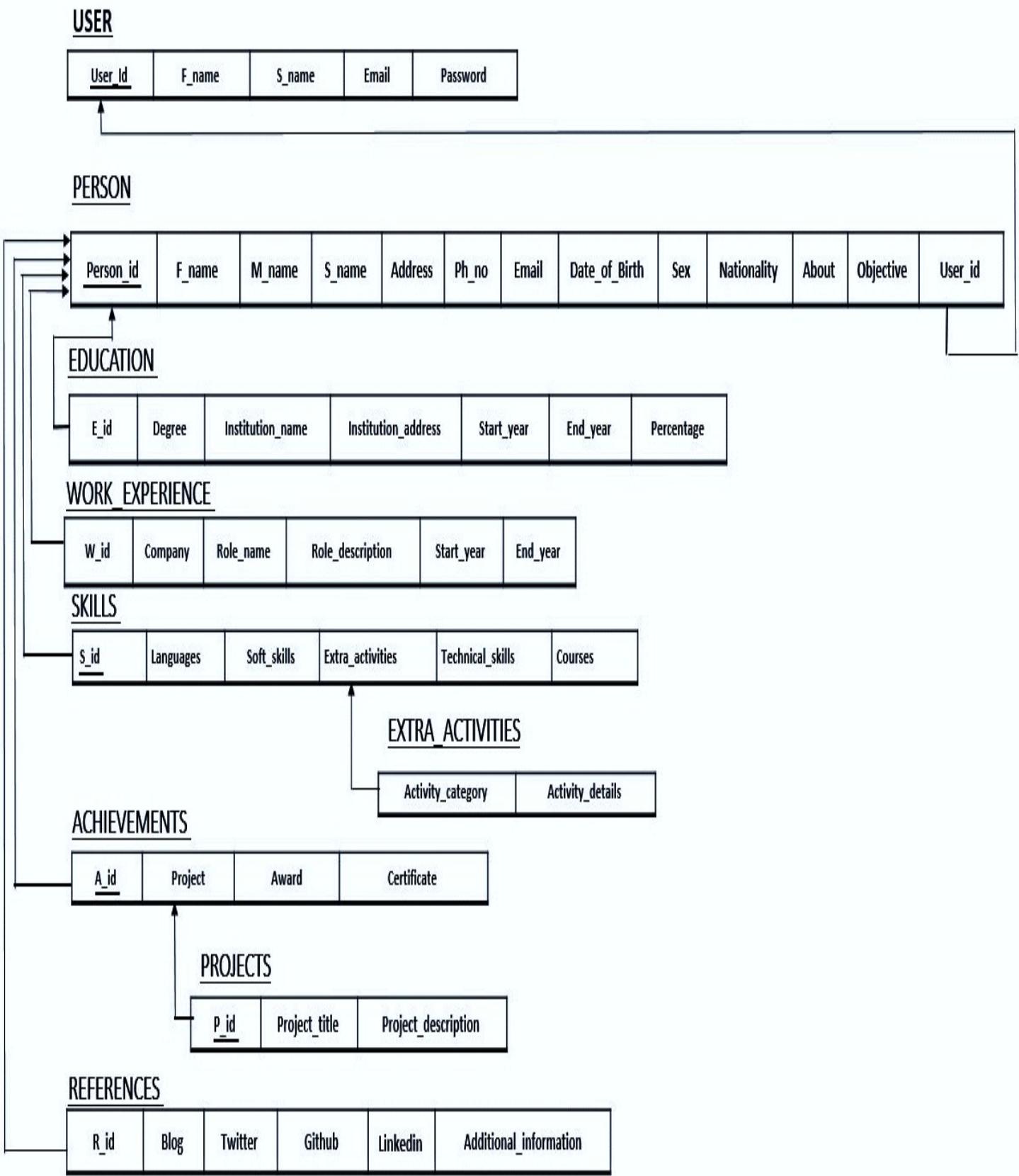


Figure 2.2 Schema Diagram

## 2.2.1 General Constraints

1. **NOT NULL:** NOT NULL constraint makes sure that a column does not hold NULL value. When we don't provide value for a particular column while inserting a record into a table, it takes NULL value by default. Most of the attributes in the project are under NOT NULL constraint.
2. **UNIQUE:** UNIQUE Constraint enforces a column or set of columns to have unique values. If a column has a unique constraint, it means that particular column cannot have duplicate values in a table. Attributes like **Email** are under this constraint.

### Key constraints

3. **PRIMARY KEY:** Primary key uniquely identifies each record in a table. It must have unique values and cannot contain nulls. Attributes like **User\_id** and **Person\_id** is under this constraint.
4. **FOREIGN KEY:** Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables. All the tables except ADMIN and PERSON are under this constraint.

## 2.2.2 Schema Description

The above Figure.2.2 shows the relational schema of Online CV Builder. It has the following entities.

- 1) **ADMIN:** This table contains the details like **User\_id**, **First name**, **Surname**, **Password** and **Email\_id** with **User\_id** as the primary key.
- 2) **PERSON:** This table contains the details of the user like **Person\_id**, **First name**, **Middle name**, **Last name**, **Address**, **Phone Number**, **Email\_id**, **Objective**, **Date of Birth**, **Nationality**, **Gender** and **About** with **Person\_id** as the primary key.
- 3) **EDUCATION:** This table contains the educational details like **Degree**, **Institution name**, **Institution address**, **Start and End Years** and **Total Percentage** with **E\_ID** referring to **Person\_id** of PERSON table.
- 4) **WORK\_EXPERIENCE:** This table contains the details of user's work experience so far like previous **company name**, **Role** and **Description** played

in previous companies and **Start and End Year** with **W\_ID** referring to **Person\_id** of PERSON table.

- 5) **SKILLS:** This table contains the details of skills acquired by the individual like **Languages known, Soft Skills, Extra activities, Technical skills and Courses** with **S\_ID** referring to **Person\_id** of PERSON table.
- 6) **EXTRA\_ACTIVITIES:** This table is a part of SKILLS table, but as it is a multivalued attribute, it is converted to a table and has details of which category the activity belongs to and its details.
- 7) **ACHIEVEMENTS:** This table contains the details of achievements of the user like **Projects, Awards and Certificate** with **A\_ID** referring to **Person\_id** of PERSON table.
- 8) **PROJECTS:** This table is a part of ACHIEVEMENTS table, but as it is a multivalued attribute, it is converted to a table and has details of the project title and its description.
- 9) **REFERENCES:** This table contains the extra information that the user would want to display like **Blog id/link, Twitter id, GitHub profile, LinkedIn profile** and other information with **R\_ID** referring to **Person\_id** of previous table.

## CHAPTER-3

# SYSTEM DESIGN

This chapter focuses on the design of the database.

### 3.1 Table Description

There are total 14 tables implemented in this project. The tables are:

- **ADMIN**
- **PERSON**
- **PERSONAL\_DETAILS**
- **EDUCATION**
- **WORK\_EXPERIENCE**
- **SKILLS**
- **EXTRA\_ACTIVITIES**
- **ACHIEVEMENTS**
- **PROJECT**
- **REFERENCES**
- **CONTACT**
- **IMAGE**
- **OPERATIONS**
- **SIGN**

Following are the contents of the table:

**ADMIN:** This table contains all the LOGIN details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	sl_no 	int(3)			No	None		AUTO_INCREMENT
2	user_id 	varchar(10)	utf8mb4_general_ci		No	None		
3	f_name	varchar(40)	utf8mb4_general_ci		No	None		
4	s_name	varchar(40)	utf8mb4_general_ci		No	None		
5	email 	varchar(50)	utf8mb4_general_ci		No	None		
6	password	varchar(40)	utf8mb4_general_ci		No	None		

*Table3.1 Admin*

**PERSON:** This table contains all the mandatory details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	sl_no 	int(3)			No	None		AUTO_INCREMENT
2	resume_id 	varchar(10)	utf8mb4_general_ci		No	None		
3	f_name	varchar(20)	utf8mb4_general_ci		No	None		
4	m_name	varchar(20)	utf8mb4_general_ci		Yes	NULL		
5	s_name	varchar(20)	utf8mb4_general_ci		No	None		
6	address	varchar(100)	utf8mb4_general_ci		No	None		
7	ph_no	bigint(15)			No	None		
8	email	varchar(50)	utf8mb4_general_ci		No	None		
9	objective	varchar(200)	utf8mb4_general_ci		No	None		
10	user_id 	varchar(10)	utf8mb4_general_ci		No	None		

*Table3.2 Person*

**PERSONAL\_DETAILS:** This table contains more details of the user.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	ID 	varchar(10)	utf8mb4_general_ci		No	None	
2	dob	date			No	None	
3	nationality	varchar(20)	utf8mb4_general_ci		No	None	
4	sex	char(1)	utf8mb4_general_ci		No	None	
5	about	text	utf8mb4_general_ci		No	None	

*Table3.3 Personal\_Details*

**EDUCATION:** This table contains the educational history of the user.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	varchar(10)	utf8mb4_general_ci		No	None		
2	grade	varchar(30)	utf8mb4_general_ci		No	None		
3	insti_name	varchar(50)	utf8mb4_general_ci		No	None		
4	insti_address	varchar(100)	utf8mb4_general_ci		No	None		
5	start_year	int(4)			No	None		
6	end_year	int(4)			No	None		
7	percentage	float			No	None		

*Table3.4 Education*

**WORK\_EXPERIENCE:** This table contains all the work-related details.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	w_id 	varchar(10)	utf8mb4_general_ci		No	None	
2	company	varchar(40)	utf8mb4_general_ci		No	None	
3	role_name	varchar(40)	utf8mb4_general_ci		No	None	
4	role_description	varchar(100)	utf8mb4_general_ci		No	None	
5	start_year	int(4)			No	None	
6	end_year	int(4)			No	None	

*Table3.5 Work\_Experience*

**SKILLS:** This table contains all the skills gained by the user.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	s_id 	varchar(10)	utf8mb4_general_ci		No	None	
2	lang_names	varchar(50)	utf8mb4_general_ci		No	None	
3	soft_skills	varchar(70)	utf8mb4_general_ci		No	None	
4	technical_skills	varchar(80)	utf8mb4_general_ci		No	None	
5	courses	varchar(100)	utf8mb4_general_ci		No	None	

*Table3.6 Skills*

**EXTRA\_ACTIVITIES:** This table contains the details of any extracurricular activities.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	a_id 	varchar(10)	utf8mb4_general_ci		No	None	
2	activity_category	varchar(70)	utf8mb4_general_ci		No	None	
3	activity_details	varchar(100)	utf8mb4_general_ci		No	None	

*Table3.7 Extra\_Activities*

**ACHIEVEMENTS:** This table contains all the achievement details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	a_id 	varchar(10)	utf8mb4_general_ci		No	None		
2	<b>certificate</b>	varchar(150)	utf8mb4_general_ci		No	None		
3	<b>award</b>	varchar(150)	utf8mb4_general_ci		No	None		

Table3.8 Achievements

**PROJECT:** This table contains project details that the user has worked on.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	a_id 	varchar(10)	utf8mb4_general_ci		No	None	
2	<b>p_no</b>	varchar(5)	utf8mb4_general_ci		No	None	
3	<b>project_name</b>	varchar(80)	utf8mb4_general_ci		No	None	
4	<b>project_description</b>	varchar(150)	utf8mb4_general_ci		No	None	

Table3.9 Project

**REFERENCES:** This table contains the extra information that user likes to provide.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	r_id 	varchar(10)	utf8mb4_general_ci		No	None	
2	<b>blog</b>	varchar(40)	utf8mb4_general_ci		No	None	
3	<b>link</b>	varchar(40)	utf8mb4_general_ci		No	None	
4	<b>twit</b>	varchar(40)	utf8mb4_general_ci		No	None	
5	<b>github</b>	varchar(40)	utf8mb4_general_ci		No	None	
6	<b>other</b>	varchar(100)	utf8mb4_general_ci		No	None	

Table3.10 References

**CONTACT:** This table provides information to contact the admin.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	user 	varchar(10)	utf8mb4_general_ci		Yes	NULL	
2	<b>messg</b>	varchar(100)	utf8mb4_general_ci		No	None	

Table3.11 Contact

**IMAGE:** This table is used to insert the profile picture for the user.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	sl_no	int(11)			No	None		AUTO_INCREMENT
2	id	varchar(10)	utf8mb4_general_ci		No	None		
3	filename	varchar(100)	utf8mb4_general_ci		No	None		

Table3.12 Image

**SIGN:** This table is used to insert the signature for the user.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id	varchar(10)	utf8mb4_general_ci		No	None	
2	filename	varchar(100)	utf8mb4_general_ci		No	None	

Table3.13 Sign

**OPERATIONS:** This table is used to implement Trigger.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	resume_id	varchar(10)	utf8mb4_general_ci		No	None	
2	action	varchar(50)	utf8mb4_general_ci		No	None	
3	date_time	datetime			No	None	

Table3.14 Operations

## 3.2 Stored Procedure

Stored Procedures are created to perform one or more DML operations on Database. It is nothing but the group of SQL statements that accepts some input in the form of parameters and performs some task and may or may not returns a value.

One stored procedure is used to identify the admin and another stored procedure is used to combine the ADMIN and PERSON table contents. This stored procedure displays the results only to the admin.

Name	Action	Type	Returns
<input type="checkbox"/> getadmin	Edit  Execute  Export  Drop	PROCEDURE	
<input type="checkbox"/> personjoinadmin	Edit  Execute  Export  Drop	PROCEDURE	

Figure 3.1 Stored Procedures

Following is the code for the stored procedure:

### 1) Stored procedure: “getadmin”

DELIMITER \$\$

CREATE DEFINER='root'@'localhost' PROCEDURE 'getadmin'()

NO SQL

SELECT \* FROM admin\$\$

DELIMITER;

### 2) Stored procedure: “personjoinadmin”

DELIMITER \$\$

CREATE DEFINER='root'@'localhost' PROCEDURE 'personjoinadmin'()

NO SQL

SELECT \* FROM PERSON AS P

INNER JOIN

admin as a where P.user\_id=a.user\_id \$\$

DELIMITER;

## 3.3 Triggers

Triggers are the SQL statements that are automatically executed when there is any change in the database. The triggers are executed in response to certain events (INSERT, UPDATE or DELETE) in a particular table. These triggers help in

maintaining the integrity of the data by changing the data of the database in a systematic fashion.

In this project triggers are added to do the following actions-

- a) On sign-up/login, irrespective of the entered case, the name of the user will be stored in database in Upper case and the email will be stored in Lower case.
- b) The above function is performed even when the user updates the values after login.
- c) A trigger which updates the Operation table, on whether the user has newly inserted the value or updated an existing value. If the user inserts a new value, INSERTED will be filled in the operations table and whereas if the existing value was updated then UPDATED will be filled in the Operations table.

<b>Name</b>	<b>Action</b>	<b>Time</b>	<b>Event</b>
<input type="checkbox"/> <b>deleteintooperation</b>	Edit  Export  Drop	BEFORE	DELETE
<input type="checkbox"/> <b>incaps</b>	Edit  Export  Drop	BEFORE	INSERT
<input type="checkbox"/> <b>insertintooperation</b>	Edit  Export  Drop	AFTER	INSERT
<input type="checkbox"/> <b>upcaps</b>	Edit  Export  Drop	BEFORE	UPDATE
<input type="checkbox"/> <b>updateintooperations</b>	Edit  Export  Drop	BEFORE	UPDATE

Figure 3.2 Triggers

Following are the Triggers used and codes for the same is mentioned-

### 1) Trigger: “**deleteintooperation**”

```
CREATE TRIGGER ‘deleteintooperation’ BEFORE DELETE ON ‘PERSON’
FOR EACH ROW INSERT INTO
OPERATIONS(PERSON_ID,ACTION,DATE_TIME)
VALUES(OLD.PERSON_ID,’DELETED’,NOW())
```

## 2) Trigger: “incaps”

```
CREATE TRIGGER ‘incaps’ BEFORE INSERT ON ‘person’  
FOR EACH ROW BEGIN  
SET NEW.f_name=UPPER(NEW.f_name);  
SET NEW.m_name=UPPER(NEW.m_name);  
SET NEW.s_name=UPPER(NEW.s_name);  
SET NEW.email=lower(NEW.email);  
END
```

## 3) Trigger: “insertintooperation”

```
CREATE TRIGGER ‘insertintooperation’ AFTER INSERT ON ‘PERSON’  
FOR EACH ROW INSERT INTO  
OPERATIONS(PERSON_ID,ACTION,DATE_TIME)  
VALUES(NEW.PERSON_ID,’INSERTED’,NOW())
```

## 4) Trigger: “upcaps”

```
CREATE TRIGGER ‘upcaps’ BEFORE UPDATE ON ‘person’  
FOR EACH ROW BEGIN  
SET NEW.f_name=UPPER(NEW.f_name);  
SET NEW.m_name=UPPER(NEW.m_name);  
SET NEW.s_name=UPPER(NEW.s_name);  
SET NEW.email=lower(NEW.email);  
END
```

**5) Trigger: “updateintooperations”**

```
CREATE TRIGGER ‘updateintooperation’ BEFORE UPDATE ON ‘PERSON’  
FOR EACH ROW INSERT INTO OPERATIONS  
(PERSON_ID,ACTION,DATE_TIME)  
VALUES (NEW.PERSON_ID,’UPDATED’,NOW())
```

## CHAPTER-4

# IMPLEMENTATION

## 4.1 Description of Tools and Technologies

This section explains about the different tool and technologies used in this project.

### Resource requirement

#### ➤ *Software requirement*

Software used in the project is as follows-

- Front End tools: HTML, JavaScript, CSS, Bootstrap
- Back End tools: PHP
- Browser that supports HTML and JavaScript
- Apache server
- MySQL database

#### ➤ *Hardware requirement*

Hardware used in the project is as follows

- CPU: Pentium processor and above
- RAM: 4 GB
- HDD: 80 GB

## 4.2 Front end development

Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that data. Front end development manages everything that users visually see first in their browser or application. Front end developers are responsible for the look and feel of a site.

## **4.2.1 Hypertext Markup Language (HTML)**

HTML is a markup language so it uses simple tags to markup and format the content. These tags are enclosed in angular brackets like <html>. Almost all the tags have their closing tags also. The <html> tag informs the browser that a HTML document has been started similarly </html> tells the end of HTML document. All the things or code written inside these two tags are sent to the browser. The browsers then display the content wrapped inside the body tag. HTML defines the format of how the web elements should be displayed. To see your content you need to save your file with .html or.htm extension and then simply run it by choosing the option to open with any browser. HTML is one of the best options to develop the webpage or website for the small or growing business that does not want to invest more in purchasing software or its license and does not require any advanced programming for their websites.

**Advantages:** HTML is Easy to Learn and Use, free, supported by all Browsers, most Friendly Search Engine and Lightweight.

## **4.2.2 Cascading style sheets (CSS)**

Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact

control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

**Advantages:** Easier to maintain and update, greater consistency in design, more formatting options, Lightweight code, faster download times and Search engine optimization benefits.

### 4.2.3 JavaScript

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

**Advantages:** Greater Speed, simple, popular, interoperable, reduces server load and creates rich interfaces.

### 4.2.4 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface

components. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps. Bootstrap includes user interface components, layouts and JS tools along with the framework for implementation. The software is available precompiled or as source code.

## 4.3 Back-end Development

Back-end development refers to the server side of an application and everything that communicates between the database and the browser. Back-end Development refers to the server side of development where you are primarily focused on how the site works. Making updates and changes in addition to monitoring functionality of the site will be your primary responsibility. This type of web development usually consists of three parts: a server, an application, and a database. Code written by back end developers is what communicates the database information to the browser. Anything you can't see easily with the eye such as databases and servers is the work of a back end developer. Back end developer positions are often called programmers or web developers.

### 4.3.1 Back-end scripting language - PHP Hypertext Preprocessor

PHP is an HTML-embedded Web scripting language. This means PHP code can be inserted into the HTML of a Web page. When a PHP page is accessed, the PHP code is read or "parsed" by the server the page resides on. The outputs from the PHP functions on the page are typically returned as HTML code, which can be read by the browser. Because the PHP code is transformed into HTML before the page is loaded, users cannot view the PHP code on a page. This make PHP pages secure enough to access databases and other secure information. A lot of the syntax of PHP is borrowed from other languages such as C, Java and Perl. However, PHP has a number of unique features and specific functions as well. The goal of the language is to allow Web developers to write dynamically generated pages quickly and easily. PHP is also great for creating database-driven Web sites.

**Advantages:** PHP has in built support for working hand in hand with MySQL, open source and free, large community document and short learning curve compared to other languages such as JSP, ASP etc.

### **4.3.2 Web Server-Apache**

An Apache Server is a web server application that delivers content such as HTML pages, multimedia and CSS Style sheets over the internet. Apache is a community-developed web application published by the Apache Software Foundation. It is arguably the most popular web server software available on the World Wide Web and is most commonly found on UNIX based operating systems such as Linux, OSX, Solaris and FreeBSD.

Apache is open source, and as such, it is developed and maintained by a large group of global volunteers. One of the key reasons Apache is so popular is that the software is free for anyone to download and use. There is no direct support or maintenance provided by the Apache foundation; however, there are vast amounts of documentation and online forums to gain help from the community. Commercial support for Apache is available from web hosting companies, such as Atlantic.Net. An Apache Server is implemented by service providers to offer clients web hosting solutions and content delivery.

Apache is a modular application, meaning that its structure allows extra functionality to be added to the core application. Apache is commonly combined with a SQL Database (such as MYSQL or Maria DB) and a server-side programming language (such as PHP).

**Advantages:** Apache can run on almost any operating systems like Windows, Linux etc., can be installed easily, highly reliable and performs better and can be modified to adjust the code and also to fix errors.

### **4.3.3 Database-MySQL**

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. MySQL is

free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability.

**Advantages:** MySQL provides Data Security, On-Demand Scalability, High Performance, Round-the-clock Uptime and the Flexibility of Open Source.

#### 4.4 User Flow Diagram

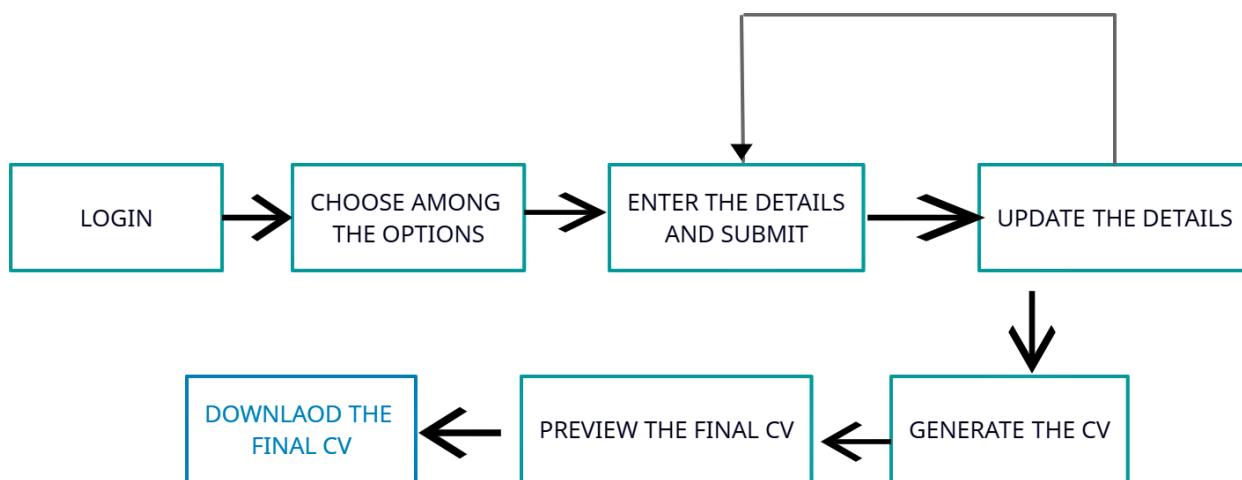


Figure 4.1 Successful CV Build

The above Figure 4.1 shows the user flow diagram for a successful CV Build. The user logs in, fills in the details and previews the CV generated and downloads his final CV in .pdf format.

#### 4.5 Discussion of Code Segment

This section explains the important code sections and modules that are implemented in the Online CV Builder project. These modules add logic to the complete system. It also talks about the integration between the front end codes and the back end MySQL database.

The following codes are implemented using HTML, CSS, JS, PHP and MySQL.

#### 4.5.1 Admin.php

```
<body>
<hr>
<div id="cont">
  <div class="form-wrap">
    <h2>ADMIN LOGIN</h2>
    <p>Access only to ADMIN.</p>
    <form class="form" action="database-admin.php" method="post">
      <div class="form-group">
        <label for="adminid">Admin_ID</label>
        <input type="text" name="adid" id="adminid" maxlength="20" required>
      </div>
      <div class="form-group">
        <label for="adpassword">Password</label>
        <input type="password" name="adpass" id="adpassword" required>
      </div>
      <button type="submit" class="btn" name="admin">Log In</button>
    </div>
  </div>
</body>
```

#### 4.5.2 Connect.php

```
<?php
function OpenCon()
{
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$db = "cv-builder";
$conn = new mysqli($dbhost, $dbuser, $dbpass,$db) or die("Connect failed: %s\n". $conn -> error);
return $conn;
}
function CloseCon($conn)
{
$conn -> close();
}
?>
```

#### 4.5.3 Contact.php

```
<body>
<div class="row">
  <div id="col1" class="col col-lg-5">
    <p class="h">CONTACT FORM</p>
```

```

<form class="form" action="database-contact.php" method="post">
  <div class="form-group">
    <label for="userid">User_ID</label>
    <input type="text" name="userid" id="userid" maxlength="10" required>
  </div>
  <div class="form-group">
    <label for="messg"> Message</label>
    <textarea type="text" name="messg" id="messg" rows="5" cols="30" required></textarea>
  </div>
  <button id="but" type="submit" class="btn">Submit</button>
</form>
<div class="other">
  <p id="p1"> <u> Contact Information</u> </p>
  <i id="i1" class="fa fa-phone-square" aria-hidden="true"></i>PH_NO: 989798989
  <br> <i id="i2" class="fas fa-envelope"></i>EMAIL_ID: online_cv_generator@gmail.com
</div>
</div>
<div id="col2" class="col col-lg-7">
  
</div>
</div>
</body>

```

#### 4.5.4 Creation.php

```

<?php
session_start();
$user=$_SESSION['user'];
?>
<body>
<a href="form.php"><button class="block1" type="button" name="button"> <p>CREATE CV
</p></button></a>
<a href="finalcv.php"> <button class="block3" type="button" name="button"><p> GENERATE
CV </p></button></a>
<a href="update.php"><button class="block1" type="button" name="button" > <p>UPDATE CV
</p></button></a>
<a href="delete.php"> <button class="block2" type="button" name="button"><p>DELETE CV
</p></button></a>
</body>

```

#### 4.5.5 Admin-database.php

```

<?php
include 'connect.php';
$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
  echo "Failed to connect to MySQL: " . $conn->connect_error;
  exit();
}

```

```

}

$rid="sup_admin_hasya";
$rpass="mar22ANDjul14";
$fid=$_POST['adid'];
$fpass=$_POST['adpass'];
if(isset($_POST['admin']))
{
    if(($fid==$rid) && ($rpass==$fpass))
    {
        ?>
        <script>
            alert("WELCOME ADMIN!!!");
        </script>
        <body>
        <hr>
        <!-- admin table -->
        <?php
            echo "<p id='pa1'>REGISTERED USERS </p>";
            echo "<table id='admin' style='width:70%; margin:auto; margin-top:70px;'><tr style='font-size:20px; font-weight:bold;'><th> USER_ID </th><th> FIRST NAME </th><th> SURNAME</th><th>EMAIL </th></tr> ";
            $sql1="SELECT * FROM admin";
            $res1=$conn->query($sql1);
            while($row=$res1->fetch_assoc())
            {
                echo"<tr><td>" . $row['user_id'] ."</td><td>" . $row['f_name'] ."</td><td>" . $row['s_name'] ."</td><td>" . $row['email'] ."</td></tr>";
            }
            echo "</table>";
            echo "<p id='pa2'> USERS AND THEIR RESUMES </p>";
            echo "<table id='tab2' style=' margin:auto; margin-top:70px; margin-bottom: 50px;'><tr style='font-size:20px; font-weight:bold;'><th> USER_ID </th><th> PERSON_ID </th><th> First Name</th><th> Middle Name</th><th> Sur_Name</th><th> Address</th><th> Ph_No</th><th> Email</th><th> Objective</th></tr>";
            $per="CALL `personjoinadmin`();";
            $fet=$conn->query($per);
            while($row1=$fet->fetch_assoc())
            {
                echo"<tr><td>" . $row1['user_id'] ."</td><td>" . $row1['Person_id'] ."</td><td>" . $row1['f_name'] ."</td><td>" . $row1['m_name'] ."</td><td>" . $row1['s_name'] ."</td><td>" . $row1['address'] ."</td><td>" . $row1['ph_no'] ."</td><td>" . $row1['email'] ."</td><td>" . $row1['objective']."'</td></tr>";
            }
            echo "</table>";
        ?>
    </body>
    <?php
}

```

```

else
{
?>
<script>
  alert("Invalid admin details.");
</script>
<?php
header("refresh:0,url=index.php");} }?>

```

#### 4.5.6 Contact-database.php

```

<?php
include 'connect.php';
// check connection
$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
  echo "Failed to connect to MySQL: " . $conn->connect_error;
  exit();
}
$user=$_POST['userid'];
$messg=$_POST['messg'];
echo $user;
$sql="INSERT into contact(user,messg) values('$user','$messg') ";
if($conn->query($sql)== True)
{
?>
<script>
  alert("MESSAGE Registered.");
</script>
<?php
header("refresh:1; url=index.php");
}
else
{
?>
<script >
  alert("ERROR: Message not Registered.");
</script>
<?php
header("refresh:1; url=contact.php");
}
?>

```

#### 4.5.7 Delete-database.php

```

<body>
<?php
include 'connect.php';

```

```

$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}
$id= $_POST['userid'];
$sql="SELECT Person_id from person where
Person_id='".$id."'";
$res=$conn->query($sql);
$name="user";
if($res->num_rows > 0)
{
    $sql2="DELETE from person where Person_id='".$id."'";
    if($conn->query($sql2)==True)
    {
        ?>
        <script type="text/javascript">
            alert("SUCCESSFULLY DELETED.");
        </script>
        <?php
        header("refresh:2; url=creation.php");
    }
    else {
        ?>
        <script type="text/javascript">
            alert("ERROR")
        </script>
        <?php
        header("refresh:2; url=delete.php");
    }
}
else
{
    ?>
    <script type="text/javascript">
        alert("INVALID RESUME ID.");
    </script>
    <?php
        header("refresh:1; url=delete.php");
    }
    ?
}
</body>

```

#### 4.5.8 Form-database.php

```
<?php
session_start();
```

```

$userid=$_SESSION['user'];
?>
<?php
error_reporting(0);
?>
<?php
function OpenCon()
{
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$db = "cv-builder";
$conn = new mysqli($dbhost, $dbuser, $dbpass,$db) or die("Connect failed: %s\n". $conn -> error);
return $conn;
}
function CloseCon($conn)
{
$conn -> close();
}
// check connection
$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno)
{
echo "Failed to connect to MySQL: " . $conn->connect_error;
exit();
}

$id= $_POST['resume'];
$exe1="SELECT Person_id from person where Person_id='$id'";
$u="SELECT Person_id from person where user_id='$userid'";
$r1=$conn->query($exe1);
$y1=$conn->query($u);
if(($r1->num_rows >0) || ($y1->num_rows >0))
{
?>
<script>
alert("Resume already created for this user, but you can update the same.")
</script>
<?php
header("refresh:1,url=update.php");
exit();
}
else
{
// person-table
$fname=$_POST['first'];
$mname=$_POST['middle'];
$surname=$_POST['sur'];

```

```

$address=$_POST['address'];
$email=$_POST['email'];
$phno=$_POST['phno'];
$objective=$_POST['objective'];
if($fname==NULL && $mname==NULL && $surname==NULL && $address==NULL &&
$email==NULL && $phno==NULL && $objective==NULL ){}
else
{
    $sql1="INSERT
person(user_id,Person_id,f_name,m_name,s_name,address,email,ph_no,objective)
values('$userid','$id','$fname','$mname','$surname','$address','$email','$phno','$objective')";
    if($conn->query($sql1)== True)
    {
    ?>
        <?php
    }
    else
    {
        ?>
        <script >
            alert("ERROR: In Personal.");
        </script>
    <?php
        header("refresh:1,url=form.php");
    }
}
//image
$filename = $_FILES["uploadfile"]["name"];
$tempname = $_FILES["uploadfile"]["tmp_name"];
$folder = "uploads/".$filename;
// Get all the submitted data from the form
$sql0 = "INSERT INTO image (id,filename) VALUES ('$id','$filename')";
if($filename==NULL)
{
}
else
{
    // Execute query
    mysqli_query($conn, $sql0);
    // Now let's move the uploaded image into the folder: image
    if(move_uploaded_file($tempname, $folder))
    {
    }
}
//signature
$signname = $_FILES["uploadsign"]["name"];
$temp = $_FILES["uploadsign"]["tmp_name"];
$folder1 = "signs/".$signname;

```

```

// Get all the submitted data from the form
$s = "INSERT INTO sign (id,filename) VALUES ('$id','$signname')";
if($signname==NULL)
{
}

} else{
// Execute query
mysqli_query($conn, $s);
// Now let's move the uploaded image into the folder: image
if(move_uploaded_file($temp, $folder1))
{
}

}
// peronel-details-table
$dob=$_POST['dob'];
$nat=$_POST['nationality'];
$sex=$_POST['gen'];
$about=$_POST['about'];
if($dob==NULL && $nat==NULL && $sex==NULL && $about==NULL)
{
}
else
{
    $sql2="INSERT into personel_details(ID,dob,nationality,sex,about)
values('$id','$dob','$nat','$sex','$about')";
    if($conn->query($sql2)== True)
    {
    }
    else
    {
        ?>
        <script>
            alert("ERROR:In personal-details.");
        </script>
        <?php
            header("refresh:1,url=form.php");
    }
}

// education
$degree=$_POST['degree'];
$iname=$_POST['iname'];
$iaaddress=$_POST['iaaddress'];
$start=$_POST['estart'];
$end=$_POST['eend'];
$cgpa=$_POST['cgpa'];
if($degree==NULL  &&  $iname==NULL  &&  $iaaddress==NULL  && $start==NULL  &&

```

```

$end==NULL && $cgpa==NULL)
{
}

else
{
    $sql3="INSERT into education(id,grade,institi_name,institi_address,start_year,end_year,percentage)
values('$id','$degree','$iname','$iaddress','$start','$end','$cgpa') ";
    if($conn->query($sql3)== True)
    {
        }

    else
    {
        ?>
        <script>
            alert("ERROR: In Education.");
        </script>
        <?php
            header("refresh:1,url=form.php");
        }
    }
}

// work_experience table
$company=$_POST['com'];
$rolen=$_POST['role'];
$roled=$_POST['desc'];
$start=$_POST['wstart'];
$end=$_POST['wend'];

if($company==NULL && $rolen==NULL && $roled==NULL && $start==NULL &&
$end==NULL)
{
}

else
{
    $sql4="INSERT into
work_experience(w_id,company,role_name,role_description,start_year,end_year)
values('$id','$company','$rolen','$roled','$start','$end') ";
    if($conn->query($sql4)== True)
    {
        }

    else
    {
        ?>
        <script>
            alert("ERROR: In WORK_EXPERIENCE.");
        </script>
    }
}

```

```

</script>

<?php
header("refresh:1,url=form.php");
}
}
// skills
$lang=$_POST['lang'];
$soft=$_POST['soft'];
$tech=$_POST['tech'];
$course=$_POST['course'];
if($lang==NULL && $soft==NULL && $tech==NULL && $course==NULL)
{
}
else
{
    $sql5="INSERT      into      skills(s_id,lang_names,soft_skills,technical_skills,courses)
values('$id','$lang','$soft','$tech','$course')";
    if($conn->query($sql5)== True)
    {
        }
    else
    {
        ?>
        <script >
            alert("ERROR: In skills.");
        </script>
        <?php
header("refresh:1,url=form.php");
    }
}

// extra-activities
$actc=$_POST['actc'];
$actd=$_POST['actd'];
if($actc==NULL && $actd==NULL)
{
}
else
{
    $sql6="INSERT into extra_activities(a_id,activity_category,activity_details)
values('$id','$actc','$actd') ";
    if($conn->query($sql6) == True)
    {
}
}

```

```

else
{
?>

<script>
alert("ERROR: In extra_activities.");
</script>
<?php
header("refresh:1,url=form.php");
}
}

// ACHIEVEMENTS
$cer=$_POST['cer'];
$award=$_POST['award'];

if($cer==NULL && $award==NULL)
{
}

else
{
    $sql7="INSERT into achievements(a_id,certificate,award) values('$id','$cer','$award') ";
    if($conn->query($sql7)== True)
    {

    }
    else
    {
        ?>
        <script>
        alert("ERROR: In Achievements");
        </script>
        <?php
        header("refresh:1,url=form.php");
    }
}

//PROJECTS
$pno=$_POST['pno'];
$pname=$_POST['name'];
$prod=$_POST['desc'];

if($pno==NULL && $pname==NULL && $prod==NULL)
{
}

else
{
    $sql8="INSERT into project(a_id,p_no,project_name,project_description)

```

```

values('$id','$pno','$pname','$prod') ";

if($conn->query($sql8)== True)
{
}

else
{
?>
<script>
alert("ERROR: In project.");
</script>
<?php
header("refresh:1,url=form.php");
}

// references
$blog=$_POST['blog'];
$link=$_POST['link'];
$twit=$_POST['twit'];
$git=$_POST['git'];
$add=$_POST['add'];

if($blog==NULL && $link==NULL && $twit==NULL && $git==NULL && $add==NULL)
{

}
else
{
    $sql9="INSERT           into          refer(r_id,blog,link,twit,github,other)
values('$id','$blog','$link','$twit','$git','$add') ";
    if($conn->query($sql9)== True)
    {

}
else
{
?>
<script>
alert("ERROR: In Refrenences.");
</script>
<?php
header("refresh:1,url=form.php");
}

?>
<script>
alert("RESUME CREATED SUCCESSFULLY!!!");
</script>

```

```

<?php
header("refresh:1,url=creation.php");
}//main-else
?>

```

#### 4.5.9 Login-database.php

```

<?php
session_start();
$_SESSION['user']=$_POST['userid'];
?>
<?php
include 'connect.php';
$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}
$id= $_POST['userid'];
$email= $_POST['email'];
$pass= $_POST['pass'];
$sql="SELECT user_id from admin where
user_id='$id' AND email='$email' AND password='$pass'";
$res=$conn->query($sql);
if($res->num_rows > 0)
{
?>
<script>
alert("SUCCESSFULLY LOGIN !!! \r\nWELCOME BACK!!!") ;
</script>
<?php
header("refresh:1; url=creation.php");
}
else {
?>
<script>
alert("INVALID Login-DETAILS");
</script>
<?php
header("refresh:1; url=loginin.php");
?
?>

```

#### 4.5.10 Signup-database.php

```

<?php
include 'connect.php';

```

```

$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}
$userid= $_POST['userid'];
$fname=$_POST['first'];
$surname=$_POST['sur'];
$email=$_POST['email'];
$pass1=$_POST['password'];
$pass2=$_POST['confirm'];
$dup = "SELECT user_id FROM admin WHERE user_id = '$userid'";
$res=$conn->query($dup);
if($res->num_rows > 0)
{
?>
<script>
    alert("Userid already exists in database. ");
</script>
<?php
    header("refresh:1; url=SignUp.php");
}
else
{
    if($pass1 == $pass2)
    {
        $sql="INSERT           into          admin(user_id,f_name,s_name,email,password)
values('$userid','$fname','$surname','$email','$pass1')";
        $insert=$conn->query($sql);
    ?>
    <script>
        alert("Successfully REGISTERED!! \r\nWELCOME TO ONLINE-CV-BUILDER FAMILY!!");
    );
    </script>

    <?php
        header("refresh:1; url=loginin.php");
    }
    else{
        ?>
        <script>
            alert("CHECK DETAILS AGAIN.");
        </script>
        <?php
            header("refresh:1; url=SignUp.php");
        ?
    }
?>

```

#### **4.5.11 Delete.php**

```
<body>
<div class="cont">
    <h1>Delete CV</h1>
    <h3>Enter valid details to DELETE your cv</h3>
    <form class="form" action="database-delete.php" method="post">
        <div class="form-group">
            <label for="userid">Person_id</label>
            <input type="text" name="userid" id="userid" maxlength="10">
        </div>
        <button id="but" type="submit" class="btn">Submit</button>
    </form>
    <br> <br>
</div>
</body>
```

#### **4.5.12 Finalcv.php**

```
<body>
<div class="cont">
    <h1 style="font-weight:bold">Generate CV</h1>
    <h3>Enter valid details to generate your cv</h3>
    <form class="form" action="template.php" method="post">
        <div class="form-group">
            <label for="userid">Person_id</label>
            <input type="text" name="userid" id="userid" maxlength="10">
        </div>
        <button id="but" type="submit" class="btn">Submit</button>
    </form>
    <br> <br>
</div>
</body>
```

#### **4.5.13 Form.php**

```
<body>
</div>
<div class="perimg">
    <p id="imgp">Upload your image here:</p>
    <input id="upload" type="file" name="uploadfile" value="" >
</div>
<div class="perimg">
    <p id="imgp">Upload your Signature here:</p>
    <input id="upload" type="file" name="uploadsing" value="" >
</div>
<div class="row">
    <div class="col1 col-lg-4">
```

```

<h3>PERSONAL</h3>
<div class="form-group1">
<label for="inputfirstname"> First Name:
  <input type="text" name="first" id="inputfirstname" maxlength="20" required>
</label>
</div>
<div class="form-group1">
<label for="inputmiddlename"> Middle Name:
  <input type="text" name="middle" id="inputmiddlename" maxlength="20" >
</label>
</div>
<div class="form-group1">
<label for="inputsurname"> Surname Name:
  <input type="text" name="sur" id="inputmiddlename" required>
</label>
</div>
<div class="form-group1">
<label for="inputaddress">Address:
  <textarea name="address" id="inputaddress" rows="3" required></textarea>
</label>
</div>
<div class="form-group1">
<label for="inputno">Phone Number:
  <input type="text" name="phno" id="inputno" maxlength="10" required>
</label>
</div>
<div class="form-group1">
<label for="inputEmail">Email Address:
  <input type="email" name="email" id="inputEmail" pattern="[a-zA-Z0-9.-_]{1,}@[a-zA-Z.-]{2, 50}{.}{1}[a-zA-Z]{2,}" required>
</label>
</div>
<div class="form-group1">
<label for="inputobjective">Objective:
  <textarea name="objective" id="inputobjective" rows="5" ></textarea>
</label>
</div>
<div class="form-group1">
<label for="inputno">Date Of Birth:
  <input type="date" name="dob" id="inputno" class="form-control" >
</label>
</div>
<div class="form-group1">
<label for="inputno">Nationality:
  <input type="text" name="nationality" id="inputno" class="form-control" maxlength="10" >
</label>
</div>
<br>

```

```

<label id="lab" for="inputsex">Sex: <br>
<input id="sex" type="radio" name="gen" value="m" >Male
<input id="sex" type="radio" name="gen" value="f" >Female
<input id="sex" type="radio" name="gen" value="o" >Others
</label>
<br>
<div class="form-group1">
    <label for="inputobjective">About:
        <textarea class="form-control" name="about" id="inputobjective" rows="10" cols="25"
></textarea>
    </label>
</div>
</div>
<div class="col2 col-lg-4">
    <h3>EDUCATION</h3>
    <div class="form-group1">
        <label for="inputdegree">Degree_Or_Grade:
            <input type="text" name="degree" id="inputdegree" >
        </label>
    </div>
    <div class="form-group1">
        <label for="inputname"> Institution_Name:
            <input type="text" name="iname" id="inputname" >
        </label>
    </div>
    <div class="form-group1">
        <label for="inputaddress"> Institution_Address:
            <input type="text" name="iaddress" id="inputaddress" >
        </label>
    </div>
    <div class="form-group1">
        <label for="inputstart"> Start_Year:
            <input type="text" name="estart" id="inputstart" maxlength="4">
        </label>
    </div>
    <div class="form-group1">
        <label for="inputend"> End_Year:
            <input type="text" name="eend" id="inputend" maxlength="4">
        </label>
    </div>
    <div class="form-group1">
        <label for="inputcgpa">CGPA/Percentage:
            <input type="text" name="cgpa" id="inputcgpa">
        </label>
        <br>
        <br>
    </div>
    <hr>
<div class="white">

```

```

<br>
<br>
</div>
<hr>
<h3>WORK_EXPERIENCE</h3>
<div class="form-group1">
  <label for="inputCOM"> COMPANY:
    <input type="text" name="com" id="inputCOM" >
  </label>
</div>
<div class="form-group1">
  <label for="inputrole"> ROLE_NAME/DESIGNATION:
    <input type="text" name="role" id="inputrole" >
  </label>
</div>
<div class="form-group1">
  <label for="inputdesc"> ROLE/DESIGNATION DESCRIPTION:
    <textarea type="text" name="desc" id="inputdesc" rows="5" cols="5"></textarea>
  </label>
</div>
<div class="form-group1">
  <label for="inputstart"> START_YEAR:
    <input type="text" name="wstart" id="inputstart" maxlength="4" >
  </label>
</div>
<div class="form-group1">
  <label for="inputend"> END_YEAR:
    <input type="text" name="wend" id="inputend" maxlength="4" >
  </label>
  <br>
  <br>
</div>
</div>
</div>
<div class="row">
  <div class="col1 col-lg-4">
    <h3>SKILLS and EXTRA_ACTIVITIES</h3>
    <div class="form-group1">
      <label for="inputlang"> SPOKEN LANGUAGES:
        <textarea type="text" name="lang" id="inputlang" rows="4" cols="6"></textarea>
      </label>
    </div>
    <div class="form-group1">
      <label for="inputsoft"> SOFT_SKILLS:
        <input type="text" name="soft" id="inputsoft" >
      </label>
    </div>
    <div class="form-group1">
      <label for="inputtechnical"> TECHNICAL_SKILLS:
        <input type="text" name="technical" id="inputtechnical" >
      </label>
    </div>
  </div>
</div>

```

```

<textarea type="text" name="tech" id="inputtechnical" rows="5"></textarea>
</label>
</div>
<div class="form-group1">
<label for="inputcourse"> COURSES:
<textarea type="text" name="course" id="inputcourse" rows="10" cols="25"></textarea>
</label>
</div>
<div class="form-group1">
<label for="inputACT"> ACTIVITY_CATEGORY:
<input type="text" name="actc" id="inputACT" >
</label>
</div>
<div class="form-group1">
<label for="inputACtd"> ACTIVITY_DESCRIPTION:
<textarea type="text" name="actd" id="inputACtd" rows="6" cols="25 "></textarea>
</label>
</div>
</div>
<div class="col2 col-lg-4">
<h3>ACHIEVEMENTS</h3>
<div class="form-group1">
<label for="inputCER"> CERTIFICATES:
<textarea type="text" name="cer" id="inputCER" rows="6" cols="25 "></textarea>
</label>
</div>
<div class="form-group1">
<label for="inputawards"> AWARDS:
<textarea type="text" name="award" id="inputawards" rows="6" cols="25 "></textarea>
</label>
<br>
<br>
</div>
<hr>
<div class="white">
<br>
<br>
</div>
<hr>
<h3>PROJECTS</h3>
<div class="form-group1">
<label for="inputpno"> PROJECT_ID:
<input type="text" name="pno" id="inputpno" >
</label>
</div>
<div class="form-group1">
<label for="inputpname">PROJECT_TITLE:
<textarea type="text" name="name" id="inputpname" rows="6" cols="15"></textarea>
</label>

```

```

</div>
<div class="form-group1">
  <label for="inputPDESC"> PROJECT_DESCRIPTION:
    <textarea type="text" name="desc" id="inputPDESC" rows="8" cols="28"></textarea>
  </label>
</div>
</div>
</div>
<div class="row">
  <div class="col3 col-lg-5">
    <h3>REFERENCES</h3>
    <div class="form-group1">
      <label for="inputblog"> BLOG_LINK:
        <input type="text" name="blog" id="inputtwit" >
      </label>
    </div>
    <div class="form-group1">
      <label for="inputlink"> LINKEDIN_ID:
        <input type="text" name="link" id="inputtwit" >
      </label>
    </div>
    <div class="form-group1">
      <label for="inputgit"> GITHUB_ID:
        <input type="text" name="git" id="inputtwit" >
      </label>
    </div>
    <div class="form-group1">
      <label for="inputtwit"> TWITTER:
        <input type="text" name="twit" id="inputtwit" >
      </label>
    </div>
    <div class="form-group1">
      <label for="inputadd"> ADDITIONAL_DETAILS:
        <textarea type="text" name="add" id="inputadd" cols="25" rows="5"></textarea>
      </label>
    </div>
  </div>
</div>
<button id="but" type="Submit" name="upload">SUBMIT</button>
<footer id="foot">
  <p>
    WARNING: This form is for new resumes i.e can't insert records into existing resumes using
    this form. To UPDATE existing resume
    <b><a href="update.php" style="text-decoration:underline" target="_parent">CLICK
    HERE</a></b> </p>
  </footer>
</body>

```

#### 4.5.14 Index.php

```
<body>
<hr>
<!-- intro -->

<div class="intro" style="margin-top:210px">
    <p id="ih">ONLINE CV BUILDER </p>
    <p class="p1">Its a platform to build unique and attractive resumes.</p>
    <p class="p2">
```

A résumé or resume is a document created and used by a person to present their background, skills, and accomplishments. Résumés can be used for a variety of reasons, but most often they are used to secure new employment.

```
    </p>
</div>
<!-- links -->
<div class="link">
    <hr>
    <div class="dots1">
        </div>
    <ul id="uli" class="nav justify-content-center">
        <li class="nav-item1">
            <a class="nav-link active" href="#types">TYPES OF RESUMES</a>
        </li>
        <li class="nav-item1">
            <a class="nav-link" href="#imp">ROLE OF RESUME</a>
        </li>
        <li class="nav-item1">
            <a class="nav-link" href="#sample">FORMAT AND SAMPLE RESUMES</a>
        </li>
        <li class="nav-item1">
            <a class="nav-link" href="#creation">BUILD CV</a>
        </li>
    </ul>
    <div class="dots2">
        </div>
    </div>
<!-- resume types -->
<section id="types">
    <br>
    <br>
    <br>
    <h2 id="th2"> <strong><U>TYPES OF RESUMES</U></strong> </h2>
```

<p class="pt2" style="width:70%; margin:auto; font-size:18px">Candidates who want to move forward in the interview process need to have a strong resume. There are two types of resumes that can make an impact with potential employers. In this section, we'll review each type so you can decide which is best for you:

```

<ul class="ul2">
    <li><a id="at2" class="nav-link" href="https://www.thebalancecareers.com/what-is-a-chronological-resume-2061944#:~:text=A%20chronological%20resume%20lists%20your,the%20list%20of%20work%20experiences." target="_blank"
        style="color:white; ">Chronological order resumes</a></li>
    <li><a id="at2" class="nav-link" href="https://resume.io/blog/customize-resume-for-each-application#:~:text=A%20custom%20resume%20can%20mean,resume%20for%20each%20job%20application.&text=In%20reality%C2%0you%20should%20be,every%20application%20you%20send%20out."
        target="_blank" style="color:white">Custom resumes</a></li>
</ul>
</p>
<br>
<h4 class="pt3">TIPS:</h4>
<ul class="ul3">
    <li>Review industry-leading examples.</li>
    <li>Apply fonts that are professional.</li>
    <li>Include relevant information.</li>
    <li>Organize by importance.</li>
    <li>Proofread and make edits.</li>
    <li>Consider active language.</li>
</ul>
</section>
<!-- importance and uses of cv --&gt;
&lt;div id="imp"&gt;
    &lt;br&gt;&lt;br&gt;
    &lt;h2 id="ih2"&gt;&lt;U&gt;ROLE OF RESUME IN EVERYDAY LIFE&lt;/U&gt;&lt;/h2&gt;
    &lt;h3 style="margin-top:30px"&gt;&lt;strong&gt;Why Is a Resume Important?&lt;/strong&gt;&lt;/h3&gt;
    &lt;p class="p1" style="width:80%; margin:auto; font-size:18px; margin-top: 10px"&gt;
        A resume is an important tool for your job search because it offers a page or two where you can display your top skills and qualities. However, a resume is much more than that. Resumes help employers make hiring decisions and help you get
        your
        first interview. That's why it matters how you structure your resume and what information you decide to include. In this article, you'll learn why a resume is important and get actionable resume tips that may help you achieve your next career
        move.
    &lt;/p&gt;
    &lt;h3 style="margin-top:30px"&gt;&lt;strong&gt; Why do employers care about your resume?&lt;/strong&gt;&lt;/h3&gt;
    &lt;p class="p2" style="width:80%; margin:auto; font-size:18px; margin-top: 10px"&gt;
        Resumes, usually accompanied by customized cover letters, get sent to employers to determine your eligibility and qualifications for a job. Employers use resumes to get a deeper understanding of candidate skills, strengths and experience.
        Your
        resume should reflect achievements, awards, education, experience and any other outstanding
    &lt;/p&gt;
</pre>

```

accomplishments that align with your career path and goals. Your resume is your first point of contact with the employer and sets the tone for

subsequent steps such as first interview, second interview, pre-screening and on-boarding.

</p>

<h3 style="margin-top:30px"> <strong>Why do you need a resume?</strong> </h3>

<p class="p3" style="width:80%; margin:auto; font-size:18px; margin-top: 10px">

In today's competitive market of professional jobs, a resume is usually a base requirement for moving forward in the interview process. A good resume will immediately display to employers why the candidate is a good fit. Here are some reasons

why you need a resume to get your next job. An effective resume:

<br>

<ul class="ull" style="font-size:18px">

<li>Outlines your relevant skills and experience</li>

<li>Displays the benefits you offer employers</li>

<li>Grabs the attention of employers</li>

<li>Matches you to the position</li>

<li>Can lead to an interview</li>

</ul>

</p>

</div>

<!-- sample resumes -->

<div id="sample">

<br>

<br>

<h1 class="h1-1">FORMATS AND SAMPLE RESUME</h1>

<div id="carouselExampleControls" class="carousel slide" data-keyboard="true" data-pause="hover">

<div class="carousel-inner data-pause=" true"">

<div class="carousel-item active">



</div>

<div class="carousel-item">



</div>

<div class="carousel-item">



</div>

<div class="carousel-item">



</div>

<div class="carousel-item">



</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

```

</a>
<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="sr-only">Next</span>
</a>
</div>
</div>
<!-- create of cv -->
<div id="creation">
  <br>
  <br>
  <div class="pic">
    <h2 style="text-align:center">Start building your CV</h2><br><i><br>
    >>>
    <button type="button" class="btn btn-warning"><a id="a1" class="nav-link" href="SignUp.php" tabindex="-1" aria-disabled="true" target="_parent"> <b>BUILD CV</b></a></button>
    <<< </div>
    <br><br>
  </div>
  <!-- icons -->
  <div id="icon">
    <i id="i1" class="fab fa-twitter fa-lg"></i> <i id="i1" class="fab fa-facebook-f fa-lg"></i> <i id="i1" class="fab fa-instagram fa-lg"></i> <i id="i1" class="fas fa-envelope fa-lg"></i>
    <p style="font-size:20px; font-weight:bold"> <i id="i2" class="fa fa-phone-square fa-lg" aria-hidden="true"></i>PHONE NUMBER: 9090909090</p>
    <p style="font-size:20px">© Copyright 2018-2020</p>
  </div>
</body>

```

#### 4.5.15 Login.php

```

<body>
<div id="cont">
  <div class="form-wrap">
    <h2>Welcome Back</h2>
    <p>Login and stay updated into the professional world.</p>
    <form class="form" action="database-login.php" method="post">
      <div class="form-group">
        <label for="userid">User_ID</label>
        <input type="text" name="userid" id="userid" maxlength="10">
      </div>
      <div class="form-group">
        <label for="email">Email</label>
        <input type="email" name="email" id="inputEmail" pattern="[a-zA-Z0-9.-_]{1,}@[a-zA-Z.-]{2,}[.]{1}[a-zA-Z]{2,40}" title="invalid mail id" required>
      </div>
      <div class="form-group">

```

```

<label for="password">Password</label>
<input type="password" name="pass" id="inputPassword" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,30}" title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" required>
    <small id="emailHelp" class="form-text text-muted"> Never share your password with anyone else.</small>
</div>
<button type="submit" class="btn">Log In</button>
<p class="mt-5 mb-3 text-muted">&copy; 2018-2020</p>
</form>
</div>
<footer>
    <p>Dont have an account? <a href="SignUp.php">Sign Up Here</a></p>
</footer>
</div>
</body>

```

#### 4.5.16 SignUp.php

```

<body>
<div id="cont">
    <div class="form-wrap">
        <h1>Sign Up</h1>
        <p>It's free and only takes a minute</p>
        <form class="form" action="database-signup.php" method="post">
            <div class="form-group">
                <label for="userid">User_ID</label>
                <input type="text" name="userid" id="userid" maxlength="10" required>
            </div>
            <div class="form-group">
                <label for="first-name">First Name</label>
                <input type="text" name="first" id="first-name" maxlength="20" required />
            </div>
            <div class="form-group">
                <label for="last-name">Last Name</label>
                <input type="text" name="sur" id="last-name" maxlength="20" />
            </div>
            <div class="form-group">
                <label for="email">Email</label>
                <input type="email" name="email" id="inputEmail" pattern="[a-zA-Z0-9.-_]{1,}@[a-zA-Z.-]{2,}[.]{1}[a-zA-Z]{2,40}" title="invalid mail id" required>
            </div>
            <div class="form-group">
                <label for="password">Password</label>
                <input type="password" name="password" id="inputPassword" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,30}" title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" required>
            </div>
            <div class="form-group">

```

```

<label for="password2">Confirm Password</label>
<input type="password" name="confirm" id="inputPassword" pattern="(?=.*\d)(?=.*[a-zA-Z])(?=.*[A-Z]).{8,30}" title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" required>
    <small id="emailHelp" class="form-text text-muted"> Never share your password with anyone else.</small>
</div>
<button type="submit" class="btn">Sign Up</button>
<p class="bottom-text">
    By clicking the Sign Up button, you agree to our
    <a href="#">Terms & Conditions</a> and
    <a href="#">Privacy Policy</a>
<p class="mt-5 mb-3 text-muted">&copy; 2018-2020</p>
</p>
</form>
</div>
<footer>
    <p>Already have an account? <a href="loginin.php">Login Here</a></p>
</footer>
</div>
</body>

```

#### 4.5.17 Template.php

```

<?php
session_start();
?>
<?php
include 'connect.php';
// check connection
$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}
$_SESSION['id']=$_POST['userid'];

$id=$_POST['userid'];
$sql="SELECT * FROM person where Person_id='$id'";
$num=$conn->query($sql);
if($num->num_rows > 0)
{
?>
<body>
<h2>CHOOSE YOUR TEMPLATE</h2>
<p> <span id="ba"></span> <span id="pr"></span> </p>
<div class="row">

```

```

<div id="col1" class="col col-lg-4">
    <h3>BASIC TEMPLATE</h3>
    <a href="template1.php">  </a>
</div>
<div id="col2" class="col col-lg-4">
    <h3>PROFESSIONAL TEMPLATE</h3>
    <a href="template2.php">  </a>
</div>
</div>
</body>
<?php
}
else
{
header("refresh:0; url=finalcv.php");
}
?>
```

#### 4.5.18 Template1.php

```

<?php
session_start();
?>
<?php
include 'connect.php';
// check connection
$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}

$id=$_SESSION['id'];
?>
<script>
    alert("YOUR FINAL CV");
</script>
<body>
    <button class="btn btn-success" id="download"> DOWNLOAD PDF</button>
    <div id="invoice">
        <!-- image -->
        <?php
        $sql0="SELECT filename from image where id='$id'";
        $res0=$conn->query("$sql0");
        if($res0->num_rows>0)
        {
            $row0=$res0->fetch_assoc();
            echo "<img id='upimg' src='uploads/" . $row0['filename'] ."' alt='person photo' width> <br>";
        }
    
```

```

else {
    echo "NO IMAGE PRESENT FOR THIS RECORD.";
}

?>
<!-- person table -->
<?php
    $sql1="SELECT * from person where Person_id='$id'";
    $res1=$conn->query($sql1);

    $sql8="SELECT * FROM refer WHERE r_id='$id'";
    $res8=$conn->query($sql8);
    $row7=$res8->fetch_assoc();
    if($res1->num_rows >0)
    {
        $row=$res1->fetch_assoc();
        echo "<b><p class='pi1'><span style='padding-left:20px'>".$row['f_name']."'</span><span style='padding-left:10px'>".$row['m_name']."'</span><span style='padding-left:10px'>".$row['s_name']."'</span></p></b>" ;
        echo "<p class='pi2'><span>".$row['email']."'</span><span style='padding-left:18px'>Linkedin: ".$row7['link']."'</span><span style='padding-left:30px'>".$row['ph_no']."'</span></p>" ;
        echo "<p class='pi3'>".$row['address']."'</p>";
        echo "<p class='pi4'>".$row['objective']."'</p>" ;
    }
    else{
        echo "NO ROWS SELECTED.";
    }
?>
<!-- personnel_details table -->
<hr>
<h2> <U>ABOUT ME: </U> </h2>
<?php
    $sql9="SELECT * from personnel_details where ID='$id'";
    $res9=$conn->query($sql9);
    if($res9->num_rows >0)
    {
        $row8=$res9->fetch_assoc();
        echo "<BR><p id='pa1'><span style='padding-right:25px;'><b style='padding-right:10px'>DATE_OF_BIRTH:</b>" ."$row8['dob']."'</span><span style='padding-right:25px'><b style='padding-right:10px;'>NATIONALITY:</b>" ."$row8['nationality']."'</span>" ;

        if($row8['sex']=='f'){
            echo "<span><b style='padding-right:10px;'>GENDER:</b> FEMALE". "</span></p>" ;
        }
        else if($row8['sex']=='m') {
            echo "<span><b style='padding-right:10px;'>GENDER:</b> MALE". "</span></p>" ;
        }
    }
}

```

```

        }
    else{
        "<span><b style='padding-right:10px;'>GENDER:</b> Others". "</SPAN></p>";
    }
    echo           "<br><h5                      STYLE='text-align:center'><b><u>BRIEF
INTRODUCTION:</u></b></h5><p id='pa2'>" . $row8['about'] ."</p>";
}
else {
    echo "NO ROWS SELECTED.";
}

?>

<!-- education-table -->
<hr>
<h2><u>EDUCATION</u></h2>
<?php
    $sql2= "SELECT * FROM education where id='$id' ORDER BY start_year";
    $res2 = $conn->query($sql2);
    if($res2 ->num_rows >0)
    {
        echo"<div style='margin-left: 40px; margin-top: 34px;'>";
        while($row1=$res2->fetch_assoc())
        {
            echo " <p class='pe1'><i id='fontawesome' class='fa fa-long-arrow-right fa-2x' aria-hidden='true'></i><b style='padding-right:15px; padding-left:20px; font-size:19px;'>DEGREE:</b><span style='padding-right:20px; font-size:17px;'>".$row1['grade']
            ."</span></p> ";
            echo "<p class='pe2' style='line-height:1.5'><b style='margin-left: 120px;font-size:18px;'><u>INSTITUTION DETAILS</u> </b><BR> <b style='padding-right:15px ;margin-left: 90px;'>INSITUTION NAME: </b><SPAN style='font-size:17px;'> " . $row1['insti_name']
            ."</SPAN><br><b style='padding-right:15px;margin-left: 90px;'> INSITUTION ADDRESS:</b><SPAN style='font-size:17px'> " . $row1['insti_address'] ."</SPAN></p>";
            echo " <p class='pe3'><b style='padding-right:15px; margin-left: 60px; font-size:19px;'>PERIOD:</b><span style='padding-right:20px'>".$row1['start_year'] ."-"
            . $row1['end_year'] ."</span></p> ";
            echo " <p class='pe4'><b style='padding-right:15px; margin-left: 60px; font-size:19px;'>PERCENTAGE/CGPA:</b><span style='padding-right:20px'>".$row1['percentage']
            ."</span></p> ";
        }
        echo "</div>";

    }
else {
    echo "NO EDUCATIONAL DETAILS.";
}

?>
<!-- work_experience -->
<hr>

```

```

<h2> <u>WORK EXPERIENCE</u></h2>
<?php
$sql3=" SELECT * FROM work_experience where w_id='".$id' ORDER BY start_year";
$res3=$conn->query($sql3);
if($res3 ->num_rows >0)
{
echo "<div style='margin-left: 40px; margin-top: 30px;'>";
while($row2=$res3->fetch_assoc())
{
echo "<p class='pe1'><i id='fontawesome' class='fa fa-long-arrow-right fa-2x' aria-hidden='true'></i><b style='padding-right:15px; padding-left:20px; font-size:19px;'>COMPANY:</b><span style='padding-right:20px; font-size:17px;'>".$row2['company']."'</span></p>";

echo "<p class='pe2' style='line-height:1.5'><b style='margin-left: 120px;font-size:18px;'><u>ROLE DETAILS:</u> </b><BR> <b style='padding-right:15px ;margin-left: 90px;'> ROLE NAME: </b><SPAN style='font-size:17px; '> " . $row2['role_name'] ."</SPAN><br><b style='padding-right:15px;margin-left: 90px;'>ROLE DESCRIPTION: </b><SPAN style='font-size:17px'>" . $row2['role_description'] ."</SPAN></p>";

echo "<p class='pe3'><b style='padding-right:15px; margin-left: 60px; font-size:19px;'>PERIOD:</b><span style='padding-right:20px'>" . $row2['start_year'] . " - " . $row2['end_year'] . "</span></p>";
}
echo "</div>";
}
else {
echo "NO Work_Experience Details.";
}
?>
<!-- skills-table -->
<hr>
<h2> <u>SKILLS</u></h2>
<?php
$sql4="SELECT * FROM skills where s_id='".$id."'";
$res4=$conn->query($sql4);

if($res4 ->num_rows >0)
{
$row3=$res4->fetch_assoc();
echo "<div style='margin-left: 100px; margin-bottom:20px; margin-top: 30px;'>";
echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>LANGUAGES:</b>".$row3['lang_names'] ."</span></p>";
echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>SOFT_SKILLS:</b>".$row3['soft_skills'] ."</span></p>";
echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>TECHNICAL_SKILLS:</b>".$row3['technical_skills'] ."</span></p>";
echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>COURSES:</b>".$row3['courses'] ."</span></p>";
echo "</div>";
}

```

```

        }
    else
    {
        echo "NO SKILLS DETAILS.";
    }
?>
<!-- extra_activities -->
<?php
$sql5="SELECT * FROM extra_activities WHERE a_id='$id'";
$res5=$conn->query($sql5);
if($res5->num_rows > 0)
{
    echo "<h4 style='text-align:center'><u>EXTRA_ACTIVITIES</u> </h4>";
    echo "<div style='margin-left: 40px; margin-top: 30px; margin-bottom: 30px;'>";
    while($row4=$res5->fetch_assoc())
    {
        echo "<br>";
        echo "<p class='pe1'><i id='fontawesome' class='fa fa-long-arrow-right fa-2x' aria-hidden='true'></i><b style='padding-right:15px; padding-left:20px; font-size:18px;'>ACTIVITY_CATEGORY:</b><span style='padding-right:20px; font-size:17px;'>" . $row4['activity_category'] . "</span></p> ";
        echo "<p class='pe3'><b style='padding-right:15px; margin-left: 60px; font-size:18px;'>ACTIVITY_DETAILS:</b><span style='padding-right:20px'>" . $row4['activity_details'] . "</span></p> ";
    }
    echo "</div>";
}
else
{
    echo "NO EXTRA_ACTIVITIES DETAILS.";
}
?>
<!-- achievements -->
<hr>
<h2> <u> ACHIEVEMENTS </u> </h2>
<?php
$sql6="SELECT * FROM achievements where a_id='$id'";
$res6=$conn->query($sql6);

if($res6->num_rows > 0)
{
    $row5 = $res6->fetch_assoc();
    echo "<div style='margin-left: 100px; margin-bottom:20px; margin-top: 30px;'>";
    echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>CERTIFICATE:</b>" . $row5['certificate'] . "</span></p> ";
    echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'> AWARDS:</b>" . $row5['award'] . "</span></p> ";
    echo "</div>";
}

```

```

else
{
    echo "NO ACHIEVEMENTS DETAILS.";
}
?>
<!-- PROJECTS -->
<?php
$sql7="SELECT * FROM project WHERE a_id='$id'";
$res7=$conn->query($sql7);
if($res7->num_rows)
{
    echo "<h4 style='text-align:center'><u>PROJECTS</u> </h4>";
    echo "<div style='margin-left: 40px ; margin-bottom: 30px; margin-top: 30px;'>";
    while($row6=$res7->fetch_assoc())
    {
        echo "<br>";
        echo "<p class='pe1'><i id='fontawesome' class='fa fa-long-arrow-right fa-2x' aria-hidden='true'></i><b style='padding-right:15px; padding-left:20px; font-size:18px;'>PROJECT ID:</b><span style='padding-right:20px; font-size:17px;'>" . $row6['p_no'] . "</span></p> ";
        echo "<p class='pe3'><b style='padding-right:15px; margin-left: 60px; font-size:18px;'>PROJECT TITLE:</b><span style='padding-right:20px'>" . $row6['project_name'] . "</span></p> ";
        echo "<p class='pe3'><b style='padding-right:15px; margin-left: 60px; font-size:18px;'>PROJECT DESCRIPTION:</b><span style='padding-right:20px'>" . $row6['project_description'] . "</span></p> ";
    }
    echo "</div>";
}
else
{
    echo "NO PROJECTS DETAILS.";
}
?>
<!-- references -->
<hr>
<h2> <u>REFERENCES </u></h2>
<?php
if($res8->num_rows)
{
    echo "<div style='margin-left: 100px; margin-bottom:50px; margin-top: 30px;'>";
    echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>BLOGS:</b>" . $row7['blog'] . "</span></p> ";
    echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>TWITTER:</b>" . $row7['twit'] . "</span></p> ";
    echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>GITHUB:</b>" . $row7['github'] . "</span></p> ";
    echo "<p class='ps1'><span><i class='fas fa-circle fa-xs'></i><b style='padding-left:6px; padding-right:15px; font-size:18px;'>ADDITIONAL DETAILS:</b>" . $row7['other'] . "</span></p> ";
    echo "</div>";
}

```

```

        }
    else
    {
        echo "NO REFERENCE DETAILS.";
    }
?>
<!-- signature -->
<?php
$sig="SELECT filename from sign where id='$id'";
$size=$conn->query("$sig");
if($size->num_rows>0)
{
    $row8=$size->fetch_assoc();
    echo " <img id='sign' src='signs/" . $row8['filename'] . "' alt='Signature photo' width> <br> ";
}
else {
    echo "NO Signature PRESENT FOR THIS RECORD.";
}
?>
</div>
</body>

```

#### 4.5.19 Template2.php

```

<?php
session_start();
?>
<?php
include 'connect.php';
// check connection
$conn = new mysqli("localhost", "root", "", "cv-builder");
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}
$id=$_SESSION['id'];
?>
<script>
    alert("YOUR FINAL CV");
</script>
<body>
<!-- pdf generate button -->
<button class="btn btn-success" id="download"> DOWNLOAD PDF</button>
    <div class="line2">
        </div>
<div id="invoice">
    <!-- image -->
    <?php

```

```

$Sql0="SELECT filename from image where id='$id'";
$res0=$conn->query("$Sql0");
if($res0->num_rows>0)
{
    $row0=$res0->fetch_assoc();
    echo " <img id='upimg' src='uploads/".$row0['filename']."' ."
alt='person photo' width><br> ";
}
else {
    echo "NO IMAGE PRESENT FOR THIS RECORD.";
}

?>

<!-- introduction -->
<?php
$Sql1="SELECT * from person where Person_id='$id'";
$res1=$conn->query($Sql1);

// reference table
$Sql8="SELECT * FROM refer WHERE r_id='$id'";
$res8=$conn->query($Sql8);
$row7=$res8->fetch_assoc();

if($res1->num_rows >0)
{
    $row=$res1->fetch_assoc();
    echo "<div class='row'><div class='col1 col-lg-4'> ."<b><p
class='p1'><span style='padding-left:5px'>" . $row['f_name'] ."</span><span style='padding-
left:5px'>" . $row['m_name'] ."</span><span style='padding-left:5px'>" . $row['s_name']
."</span></p></b>";
        echo "<p class='p12'>" . $row['objective'] ."</p></div>";
        echo "<div class='col2 col-lg-5'><i class='fas fa-envelope fa-sm'
style='padding-right:5px'></i>" . $row['email'] ."<br><i class='fab fa-linkedin fa-sm' style='padding-
right:7px'></i>" . $row7['link'] ."<br><i class='fas fa-phone-alt fa-sm' style='padding-
right:7px'></i>" . $row['ph_no'];
        echo "</div></div>";
}
else
{
    echo "NO DATA present.";
}
?>
<div class="line">
</div>
<!-- about -->
<h4 style="padding-left:90px"> SHORT ASIDE </h4>
<hr>

```

```

<?php
$sql9="SELECT * from personel_details where ID='$id'";
$res9=$conn->query($sql9);
if($res9->num_rows >0)
{
    $row8=$res9->fetch_assoc();
    echo "<p id='pa1'>" . $row8['about'] . "</p>";
}
else
{
    echo "NO DATA present.";
}
?>
<!-- education -->
<h4 style="padding-left:90px"> EDUCATION HISTORY </h4>
<hr>
<?php
$sql2= "SELECT * FROM education where id='$id' ORDER BY start_year";
$res2 = $conn->query($sql2);
if($res2 ->num_rows >0)
{
    while($row1=$res2->fetch_assoc())
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span>" . $row1['start_year'] . "</span>-<span>" . $row1['end_year'] . "</span></div>";
        echo "<div class='col4 col-lg-6'>" . $row1['grade'] . "<br>" .
$row1['insti_name'] . "<br>" . $row1['insti_address'] . "<br><span style='padding-right:5px'>SCORE:</span>" . $row1['percentage'] . "</div></div>";
    }
}
else
{
    echo "NO DATA present.";
}
?>
<!-- work_experience -->
<h4 style="padding-left:90px"> EMPLOYMENT HISTORY </h4>
<hr>
<?php
$sql3=" SELECT * FROM work_experience where w_id='$id' ORDER BY
start_year";
$res3=$conn->query($sql3);
if($res3 ->num_rows >0)
{
    while($row2=$res3->fetch_assoc())
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span>" . $row2['start_year'] . "</span>-<span>" . $row2['end_year'] . "</span></div>";
        echo "<div class='col4 col-lg-6'>" . $row2['company'] . "<br>"
```



```

        echo      "<div      class='col4      col-lg-6'><span>" . $row3['technical_skills'] ." </span></div></div>";
    }
    if($row3['courses'] != NULL)
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span> COURSES: </span></div>";
        echo "<div class='col4 col-lg-6'><span> " . $row3['courses'] . "
</span></div></div>";
    }
}
else
{
    echo "NO DATA present.";
}
?>
?>
<!-- additional information -->
<h4 style="padding-left:90px"> ADDITIONAL INFORMATION </h4>
<hr>
<?php
$sql5="SELECT * FROM extra_activities WHERE a_id='$id'";
$res5=$conn->query($sql5);
if($res5->num_rows > 0)
{
    $text=explode(" ",$row3['lang_names']);
    echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span> LANGUAGES-SPOKEN: </span></div>";
    echo "<div class='col4 col-lg-6'><span>";
    foreach($text as $re)
    {
        echo $re, '<br>';
    }
    echo "</span></div></div>";
    while($row4=$res5->fetch_assoc())
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span> CATEGORY: </span></div>";
        echo "<div class='col4 col-lg-6'>" . $row4['activity_category'] . "<br><br><span> " . $row4['activity_details'] . " </span></div></div>";
    }
}
else
{
    echo "NO DATA present.";
}
?>
<!-- reference -->
<h4 style="padding-left:90px"> REFERENCES </h4>

```

```

<hr>
<?php
if($res8->num_rows)
{
    if($row7['blog'] != NULL)
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span>BLOGS: </span></div>";
        echo "<div class='col4 col-lg-6'><span> " . $row7['blog'] . "
</span></div></div>";
    }
    if($row7['twit'] != NULL)
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span>TWITTER: </span></div>";
        echo "<div class='col4 col-lg-6'><span> " . $row7['twit'] . "
</span></div></div>";
    }
    if($row7['github'] != NULL)
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span>GITHUB: </span></div>";
        echo "<div class='col4 col-lg-6'><span> " . $row7['github'] . "
</span></div></div>";
    }
    if($row7['other'] != NULL)
    {
        echo "<div id='row1' class='row'><div class='col3 col-lg-3'>
<span>OTHERS: </span></div>";
        echo "<div class='col4 col-lg-6'><span> " . $row7['other'] . "
</span></div></div>";
    }
}
else
{
    echo "NO DATA present.";
}

?>

<!-- signature -->

<?php

```

```

$sig="SELECT filename from sign where id='$id'";
$sige=$conn->query("$sig");
if($sige->num_rows>0)
{

```

```

        $row8=$sige->fetch_assoc();
        echo "<img id='sign' src='signs/" . $row8['filename'] ."' alt='Signature
photo' width><br>";
    }
    else {
        echo "NO Signature PRESENT FOR THIS RECORD.";
    }

?>
<div class="line3">
</div>
</div>
</body>

```

#### 4.5.20 Update.php

```

<body>
<div class="row">
<div id="col" class="column col-lg-3">
<a href="update-tables/person.php" target="_parent">PERSONAL_DETAILS</a>
</div>
<div id="col" class="COLUMN col-lg-3">
<a href="update-tables/education.php" target="_parent">EDUCATION</a>
</div>
<div id="col" class="column col-lg-3">
<a href="update-tables/work.php" target="_parent">WORK_EXPERIENCE</a>
</div>
</div>
<div class="row">
<div id="col" class="column col-lg-3">
<a href="update-tables/skill.php" target="_parent">SKILLS </a>
</div>
<div id="col" class="COLUMN col-lg-3">
<a href="update-tables/extra.php" target="_parent">EXTRA_ACTIVITIES</a>
</div>
<div id="col" class="column col-lg-3">
<a href="update-tables/achievement.php" target="_parent">ACHIEVEMENTS</a>
</div>
</div>
<div class="row" background-color="#3797a4">
<div id="col2" class="column col-lg-4">
<a href="update-tables/project.php" target="_parent">PROJECTS</a>
</div>
<div id="col2" class="column col-lg-4">
<a href="update-tables/reference.php" target="_parent">REFERENCES</a>
</div>
</div>
</body>

```

## 4.6 Discussion of Results

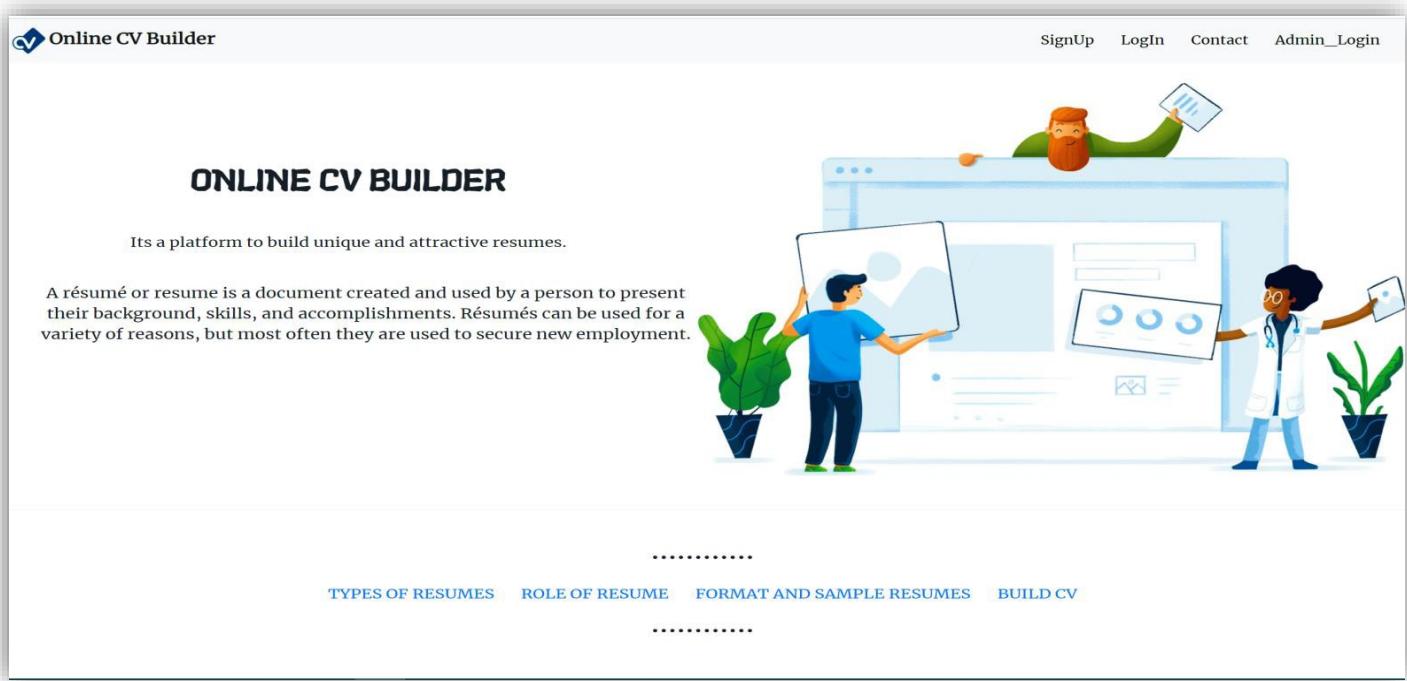


Figure 4.2 Homepage

The above Figure 4.2 is the snapshot of the homepage with various links. The user can choose to either Sign up, Login or Contact the Admin. The User can read about the types of resumes, role of resume and go through the sample CVs provided.

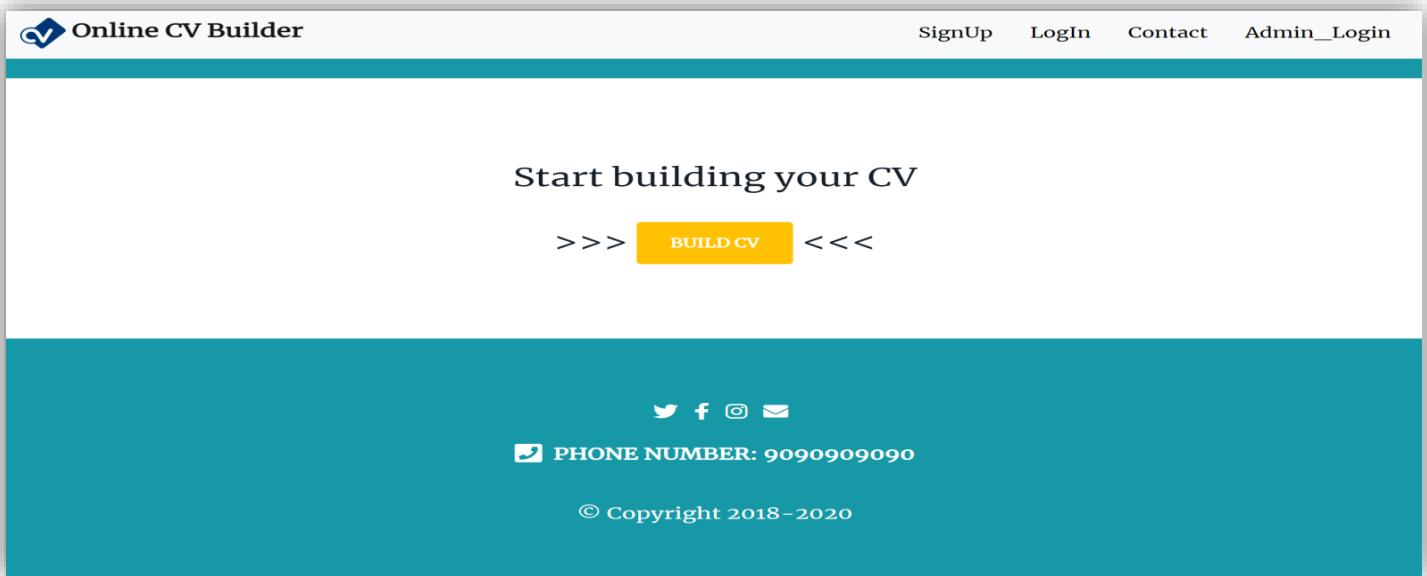
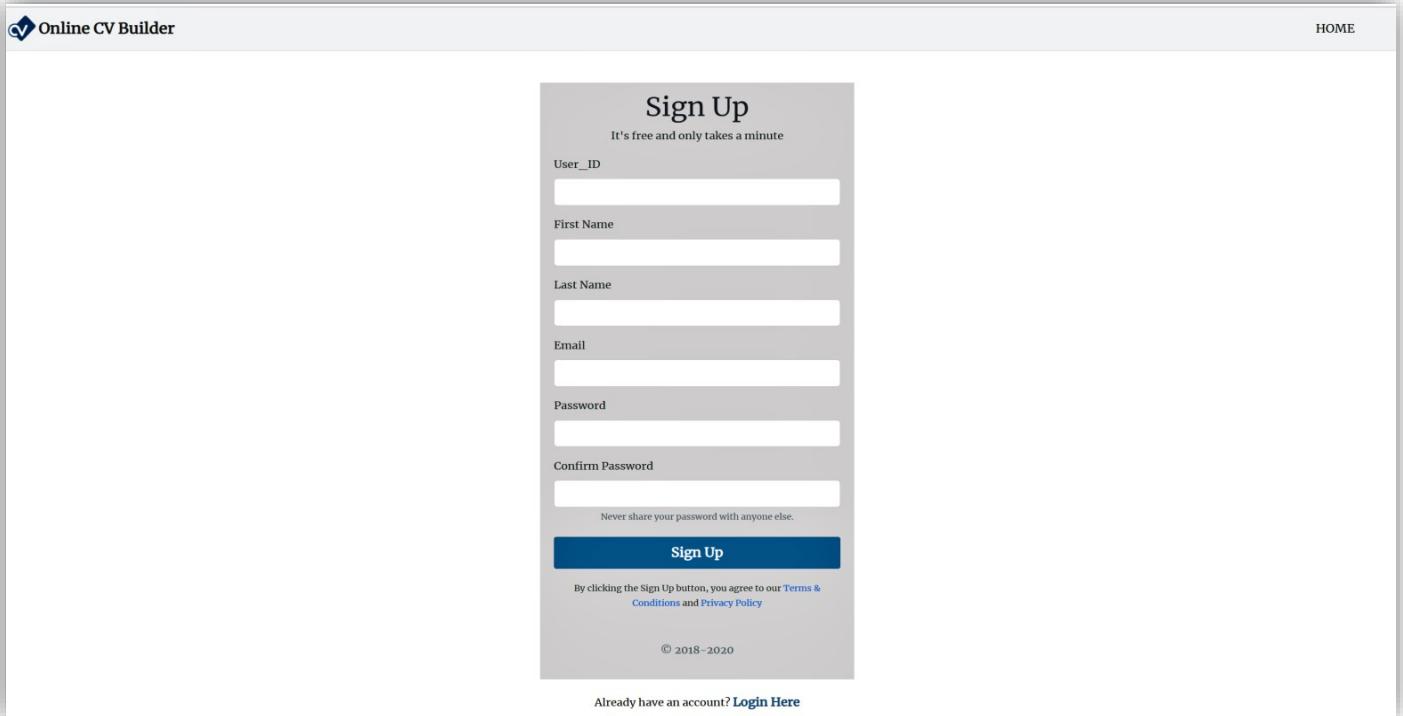


Figure 4.3 Homepage with Create CV link

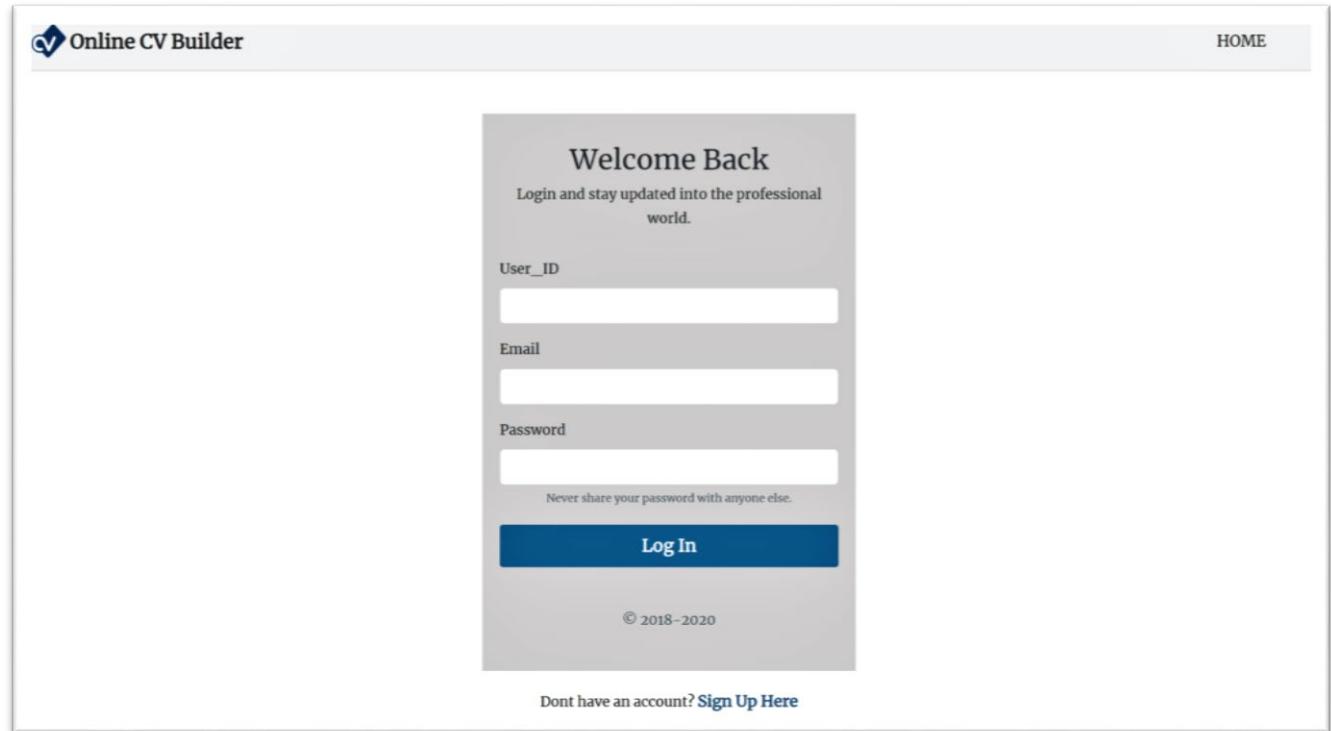
The above Figure 4.3 is the snapshot of the homepage where a button is provided which will link to the Sign up or Login prompt.



The screenshot shows the 'Sign Up' page of the Online CV Builder website. At the top, there's a header with the logo 'Online CV Builder' and a 'HOME' link. Below the header is a large central box with a light gray background. The title 'Sign Up' is at the top of this box, followed by the subtext 'It's free and only takes a minute'. There are six input fields labeled 'User\_ID', 'First Name', 'Last Name', 'Email', 'Password', and 'Confirm Password'. Below these fields is a note: 'Never share your password with anyone else.' A large blue 'Sign Up' button is centered below the note. At the bottom of the box, a small note states: 'By clicking the Sign Up button, you agree to our [Terms & Conditions](#) and [Privacy Policy](#)'. The footer of the page includes the copyright notice '© 2018–2020' and a link 'Already have an account? [Login Here](#)'.

*Figure 4.4 Sign-up prompt*

The above Figure 4.4 is the snapshot of the Sign Up prompt, where the user has to fill in his user\_id, Name, Email and create a Password. Only valid Password is accepted.



The screenshot shows the 'Welcome Back' login page of the Online CV Builder website. At the top, there's a header with the logo 'Online CV Builder' and a 'HOME' link. Below the header is a large central box with a light gray background. The title 'Welcome Back' is at the top of this box, followed by the subtext 'Login and stay updated into the professional world.'. There are three input fields labeled 'User\_ID', 'Email', and 'Password'. Below the 'Password' field is a note: 'Never share your password with anyone else.' A large blue 'Log In' button is centered below the note. The footer of the page includes the copyright notice '© 2018–2020' and a link 'Dont have an account? [Sign Up Here](#)'.

*Figure 4.5 Login Prompt for Existing users*

The above Figure 4.5 is the snapshot of the Login prompt for an existing user. A user can sign up only once with same name and email id. The user has to fill in the User\_id, Email and Correct Password.

*Figure 4.6 Login Prompt for Admin*

The above Figure 4.6 is the snapshot of the Login prompt for Admin, who can log in through his ID and password.

**REGISTERED USERS**

USER_ID	FIRST NAME	SURNAME	EMAIL
123	abc	abc	abc@gmail.com
abc123	shruthi	kumar	shruthi@gmail.com
kumar123	kumar	singh	kumar@gmail.com
shreya01	Sushant	Alavala	supriya.sportstar@gmail.com
shreya123	shreya	uday hasyagar	shreya.hasyagar@gmail.com
suppi	Supriya	Alavala	supriya.sportstar@gmail.com

**USERS AND THEIR RESUMES**

USER_ID	RESUME_ID	First_Name	Middle_Name	Sur_Name	Address	Ph_No	Email	Objective
suppi	1201	Supriya		Alavala	158-33 1ST MAIN,3RD CROSS,CANARA BANK COLONY CHIKKALASANDRA	2147483647	supriya.sportstar@gmail.com	
shreya123	shreya01	shreya	UDAY	uday hasyagar	Bangalore	9848586899	shreya.hasyagar@gmail.com	cyber security.

*Figure 4.7 Result of Trigger*

The above Figure 4.7 is the snapshot of the registered users available only for the Admin. This is the result of the Triggers discussed and applied in the section 3.3

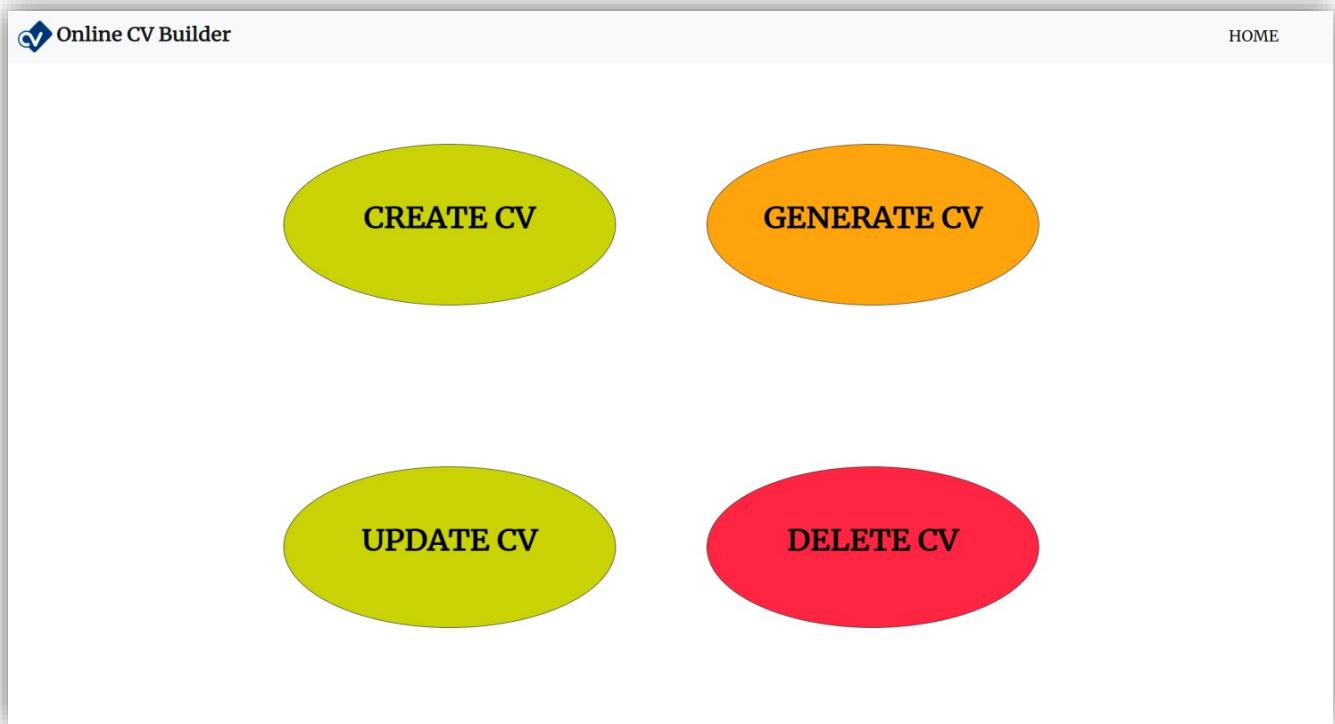


Figure 4.8 Options for the User

The above Figure 4.8 is the snapshot of the Options that the user is free to choose once logged in. The user can create CV by filling in the details about him and generate CV. The user can also modify the fields of his CV or delete his existing CV.

A screenshot of a "RESUME-DETAILS" form. At the top center is a blue diamond-shaped logo with a white "CV" monogram. Below it is the title "RESUME-DETAILS" in bold green capital letters. A text input field labeled "Resume\_ID:" is followed by a blank input box. Below this is a section for uploading an image, with the text "Upload your image here:" and a "Choose File" button showing "No file chosen". Another section for uploading a signature follows, with the same text and button. At the bottom, there are two large grey rectangular sections. The left section is labeled "PERSONAL" and contains a "First Name:" label and a text input field. The right section is labeled "EDUCATION" and contains a "Degree\_Or\_Grade:" label and a text input field.

Figure 4.9 Form to fill in necessary details

The above Figure 4.9 is the snapshot of Create CV. The user will have to enter his Person\_id, upload his photograph and signature and enter his Personal Details, Educational history, Work Experiences, Skills acquired, Projects worked on, Achievements and Additional Details.

**REFERENCES**

BLOG\_LINK:

LINKEDIN\_ID:

GITHUB\_ID:

TWITTER:

ADDITIONAL\_DETAILS:

**SUBMIT**

WARNING: This form is for new resumes i.e can't insert records into existing resumes using this form. To UPDATE

*Figure 4.10 Submission on filling mandatory field*

The above Figure 4.10 is snapshot of the page where the user can enter his details and click Submit. If the mandatory fields are missed, the user will be notified and only then accepted. The submitted details will be stored in the database.



*Figure 4.11 Fields to choose for update*

The above Figure 4.11 is the snapshot of the fields that the user can choose to change existing details.

The screenshot shows a form titled "EDUCATION-DETAILS". At the top center is a logo consisting of a blue "Q" inside a white square. Below the logo are seven input fields with labels: "RESUME\_ID:", "DEGREE\_OR\_GRADE:", "INSTITUTION\_NAME:", "INSTITUTION\_ADDRESS:", "START\_YEAR:", "END\_YEAR:", and "CGPA/PERCENTAGE:". A large blue "UPDATE" button is located at the bottom of the form area.

Figure 4.12 Page is as shown if user decides to update Education field

The above Figure 4.12 is the snapshot of Education field if that was the field opted by the user to update. On clicking Update Button, the details will be updated in the database.

The screenshot shows a form titled "REFERENCES- DETAILS". At the top center is a logo consisting of a blue "Q" inside a white square. Below the logo are six input fields with labels: "RESUME\_ID:", "BLOG\_LINK:", "LINKEDIN\_ID:", "GITHUB\_ID:", "TWITTER:", and "ADDITIONAL\_DETAILS:". A navigation bar at the top includes links for "HOME", "CATEGORY", and "BUILD-CV".

Figure 4.13 Page is as shown if user decides to update References field

The above Figure 4.13 is the snapshot of References field if the user decides to update his additional information. Any number of specified fields can be updated, without limitation on modification.

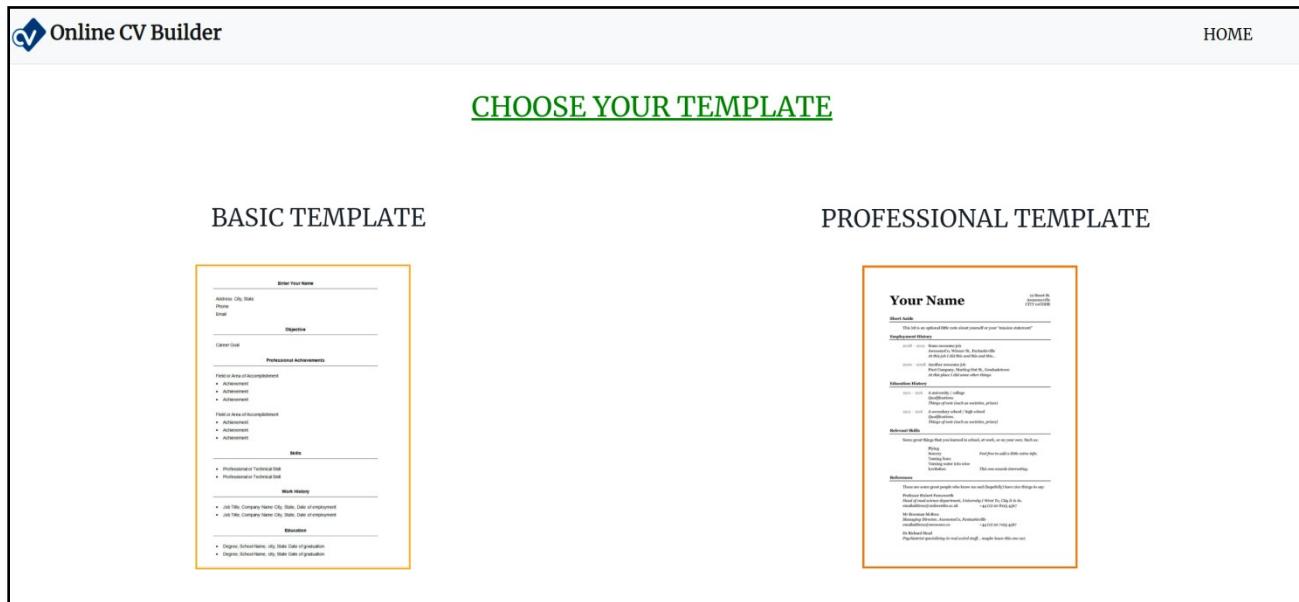


Figure 4.14 Options for Templates

The above Figure 4.14 is the snapshot of the options of template available for the user to choose from. The user can choose predefined format to create his CV.

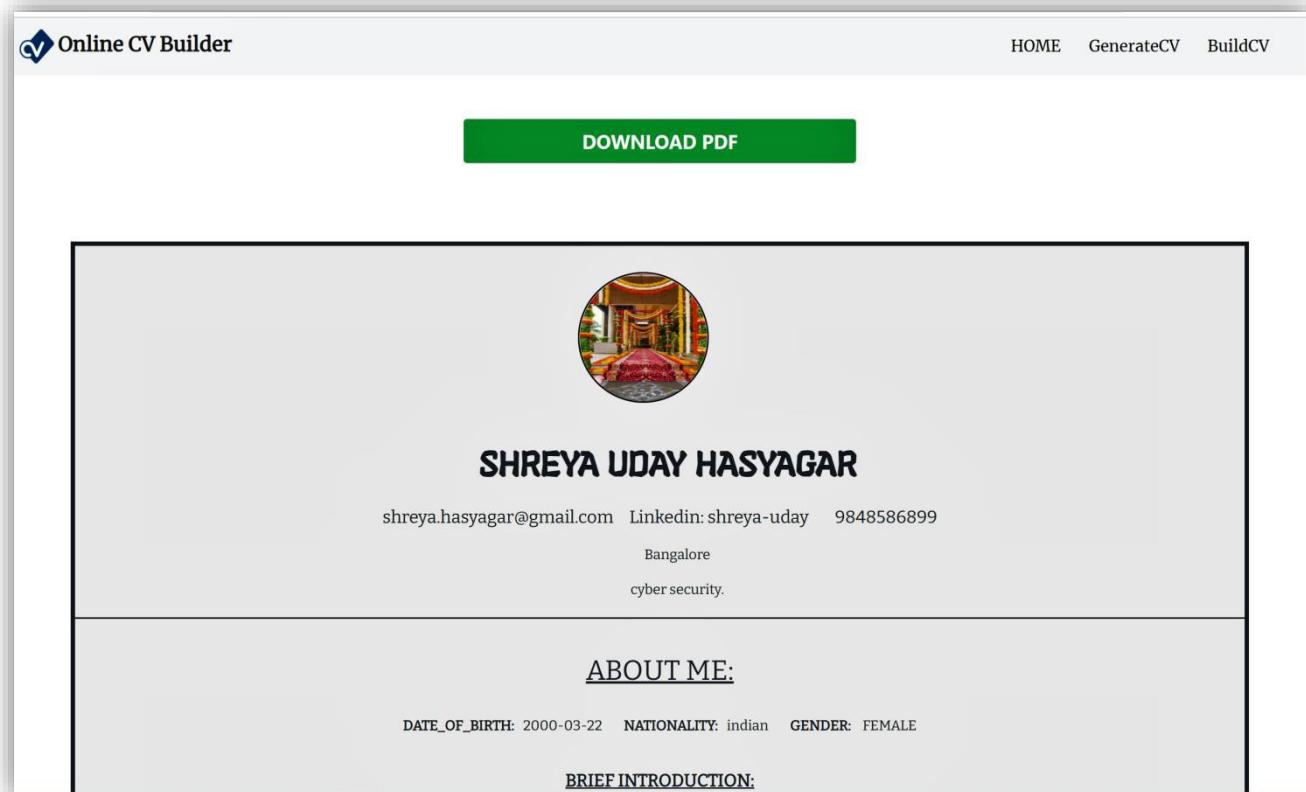


Figure 4.15 a) Output for Basic Template

<ul style="list-style-type: none"> <li>● LANGUAGES: english kannada</li> <li>● SOFT_SKILLS: speaking</li> <li>● TECHNICAL_SKILLS: java c c++</li> <li>● COURSES: java c c++</li> </ul> <p style="text-align: right;"><u><a href="#">EXTRA ACTIVITIES</a></u></p> <p>→ ACTIVITY_CATEGORY: dance ACTIVITY_DETAILS: kathakali</p> <p>→ ACTIVITY_CATEGORY: sports ACTIVITY_DETAILS: throwball badminton chess ludo</p>	<p style="text-align: center;"><u><a href="#">ACHIEVEMENTS</a></u></p> <ul style="list-style-type: none"> <li>● CERTIFICATE: singing badminton chess</li> <li>● AWARDS: java c c++ python</li> </ul> <p style="text-align: right;"><u><a href="#">PROJECTS</a></u></p> <p>→ PROJECT_ID: aaf PROJECT_TITLE: app development PROJECT_DESCRIPTION: using java</p> <p>→ PROJECT_ID: p1 PROJECT_TITLE: dbms PROJECT_DESCRIPTION: event management</p>
<p style="text-align: center;"><u><a href="#">REFERENCES</a></u></p> <ul style="list-style-type: none"> <li>● BLOGS: shreya01@security</li> <li>● TWITTER:</li> <li>● GITHUB: shreya.hasyagar</li> <li>● ADDITIONAL DETAILS: fb: shreya.hasyagar</li> </ul> <p style="text-align: center;"></p>	

Figure 4.15 b) Output for Basic Template

The above Figure 4.15 is the snapshot is a preview of a filled CV when the user chooses Basic Template. On clicking “Download PDF” button, Final CV will be automatically downloaded.

 Online CV Builder
[HOME](#)
[GenerateCV](#)
[BuildCV](#)

---

[DOWNLOAD PDF](#)

---



**SHREYA UDAY HASYAGAR**

cyber security.

✉ shreya.hasyagar@gmail.com  
✉ shreya-uday  
📞 9848586899

---

**SHORT ASIDE**

---

Studying in RNSIT. Branch Information Science.

**EDUCATION HISTORY**

---

2018-2022	btech	rns
-----------	-------	-----

Figure 4.16 a) Output for Professional CV Template

2000-2005	IBM JAVA DEVELOPER not a java developer
2000-2002	asif asifali igdig
2005-2005	IBM Web Developer asifali
<b>PROJECT WORK</b>	
ad	app development using java
pi	dbms event management
<b>RELEVANT SKILLS</b>	
SOFT-SKILLS	speaking
TECHNICAL-SKILLS	java c++
COURSES	java c++
<b>ADDITIONAL INFORMATION</b>	
LANGUAGES-SPOKEN:	english kannada
CATEGORY:	dance kathakali
CATEGORY:	sports throwball badminton chess ludo
<b>REFERENCES</b>	
BLOGS	shreyashisecurity
GITHUB	shreya_hayagar
OTHERS	fb shreya.hayagar

*[Handwritten signature]*

Figure 4.16 b) Output for Professional CV Template

The above Figure 4.16 is the snapshot of the preview of a filled CV when the user chooses Professional CV template. On clicking “Download PDF” button, Final CV will be downloaded automatically.

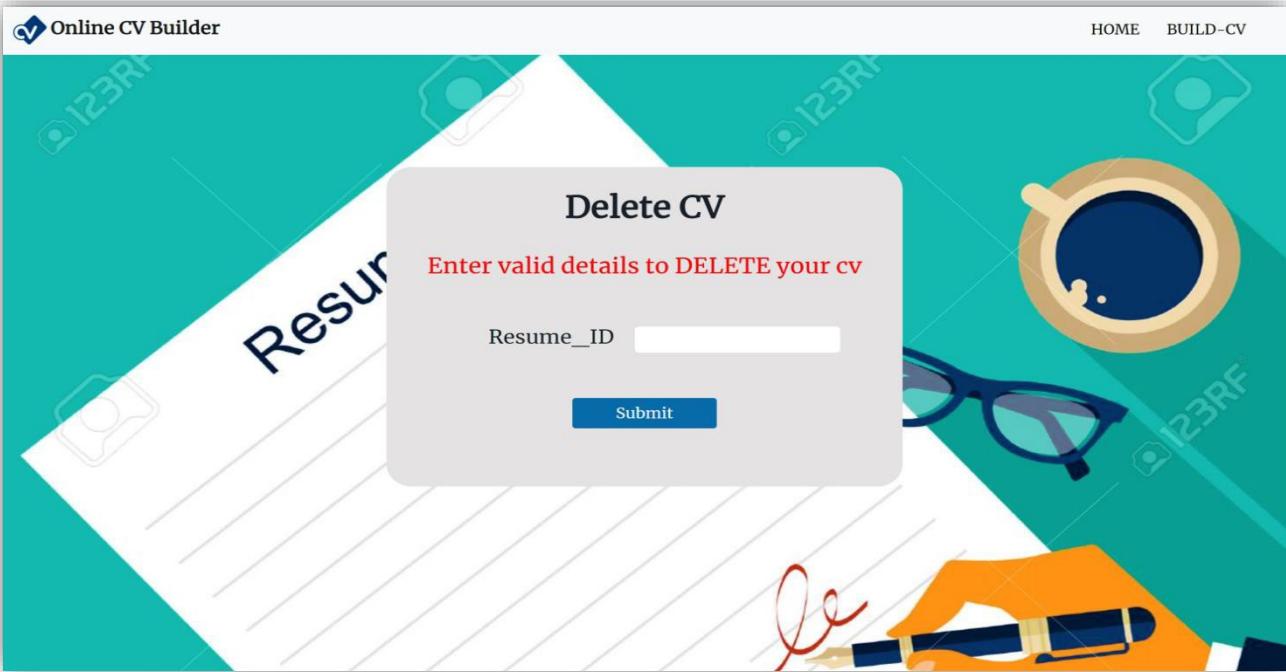


Figure 4.17 Delete CV

The above Figure 4.17 is the snapshot of the Delete CV prompt when the user chooses to delete an existing CV. CV will not be deleted if the Person\_id is not registered.

## 4.7 Applications of the Project

- Helps in preparing a quality CV when there is little time to do so.
- Provides comfortable and smooth interface to fill in the details.
- Provides options to pick between different templates.
- The project formats all the information and also provides an example of how the CV should look.
- The project does not allow the user to miss out to fill in the important fields.
- Helps to get a quality CV for free instead of spending hundreds of rupees on it.
- Helps in landing more interviews simply by creating a CV which helps you to stand out of the crowd.
- Provides Data Security and also helps in downloading a sharable document.

## **CHAPTER-5**

# **CONCLUSIONS AND FUTURE ENHANCEMENTS**

### **5.1 Conclusion**

This project was an attempt to make the structure and working of an Online CV Builder which is user-friendly. This was an attempt to make it similar to the real world implementation. In this project, all the necessary fields that are needed to build a quality CV was achieved in a constructive manner. Given the right guidance and support its applications and availability can be enhanced.

### **5.2 Future Enhancements**

- The project can be enhanced to manage multiple CV versions.
- Email and message can be sent to the user reminding to keep making changes to their CV in order to compete with the world, based on the last update.
- Providing a Live Feedback by content Optimizer to analyze the CV and suggest revisions to increase the quality of the user's details.
- Adding more attractive templates and allowing the user to customize the Final downloadable CV.
- Adding articles and notifying the user through Email regarding the job opening based on the skills of the user.
- Recovery of Person\_id and Password can be provided when the user forgets either password or Person\_id.

## REFERENCES

- <https://www.w3schools.com/>
- <https://www.wikipedia.org/>
- <https://www.tutorialspoint.com/index.htm>
- <https://www.apachebooster.com/>
- <https://www.atlanticnet.com/>
- <https://www.whatis.techtarget.com/>
- <https://www.geeksforgeeks.org/>
- Fundamentals of Database Systems, Ramez Elmasri and Shamkant B.Navarthe, 7<sup>th</sup> Edition, 2017, Pearson
- Database management systems, Ramakrishnan and Gehrke, 3<sup>rd</sup> Edition, 2014, McGraw Hill