

Facile Hire System

Fundamentals of Database Systems
Computer Science Department
University of North Texas

Asmaa Alghamdi

Vaishnavi Dittakavi

Manohar Kowthavarapu

Thoa Nguyen

Aima Ovai

Nayana Shrestha

Table of Contents

1. Facile Hire System.....	4
1.1 Project Description	4
1.2 Assumptions.....	4
1.3 Required Technology	4
1.4 E-R Model (see attached file)	5
2. Normalized Data Structure Tables, Relational Schema, and SQL Statements	5
2.1. Relational Schema	5
2.2 Relational Schema Diagram (see attached file).....	6
2.3 SQL Statements.....	6
3. Sample Data and UI Component	16
3.1.1 Sample Data Loaded	16
3.1.2 Functions and Queries	19
3.2 UI Component.....	22
3.2.1 Homepage.....	22
3.2.2 Candidate Portal	22
3.2.2.1 Logging into the candidate portal	22
3.2.2.2 Viewing the Candidate's dashboard.....	23
3.2.2.3 Viewing the candidate's profile/resume	23
3.2.2.4 Logging out of the candidate portal	24
3.2.3 Recruiter Portal.....	24
3.2.3.1 Logging into the recruiter portal	24
3.2.3.2 Viewing the recruiter Dashboard	25
3.2.3.3 Viewing the recruiter Jobs page	26
3.2.3.4 Uploading Resumes to Recruiter Portal	26
3.2.3.5 Uploading Jobs on Recruiter Portal	28
3.2.3.6 Viewing Candidates from Recruiter Portal	31
3.2.3.7 Viewing the Recruiter's Profile	32
3.2.3.8 Logging out of recruiter profile	33
3.2.4 Employee Portal.....	33
3.2.4.1 Logging into the Employee Portal.....	33
3.2.4.2 The Employee Portal Dashboard	34
3.2.4.3 Viewing a candidate's profile status from employee portal	35

3.2.4.4 Logging out of employee portal	36
4. Appendix	36
5. References	36

1. Facile Hire System

1.1 Project Description

"Parsing Candidate Resumes, Recruiting Candidates, and Maintaining Interview Details" is the system we propose. There are three main sorts of logins available on our platform:

1. Recruiter login into his portal and uploads the resumes that he or she has. The resume parser service will parse those resumes and collect the candidates' information, which will then be stored in a database. The parsed candidates will appear in the UI for the appropriate recruiter. Now the recruiter contacts the candidate and begins updating all of the candidate's information.
2. Employee login Employee login into his portal and searches for the candidates he wishes to interview; once the interview is completed, he/she updates the candidates on the progress of their interview.
3. Candidate login. Candidates can log in to check their interview status, BGC status, and other information.

1.2 Assumptions

1. A recruiter can only work for one company at a time.
2. The recruiter logs in to the portal and uploads candidates resumes.
3. The resumes that are uploaded will be processed and stored in various tables. Resume parser will extract and insert information to candidates, experience, education, skills, certifications and projects.
4. Recruiter can post the jobs.
5. The recruiter will select one open position and identify applicants from the candidate dashboard, continuing to call candidates who are qualified for the position.
6. The candidate's call status is updated by the recruiter. If an applicant is interested in the position, he/she will set up an interview.
7. If a candidate is interested in the job position, the recruiter will schedule the interview.
8. Employee will conduct the interview and update the status of the interview using the employee portal.
9. If the candidate is chosen, the interview status will be changed to selected, and the recruiter will begin the background check. Additionally, by logging onto the candidate site, the candidate can view the company's advantages and background check status.
10. During the background verification process, the recruiter requests BGC papers from the candidates. Once the candidate submits the BGC documents, the recruiter updates the portal with all of the documents submitted, which the candidate can view in his portal.
11. Throughout the BGC process, the recruiter updates the status of the BGC as Initiated, Processing, and Completed.

1.3 Required Technology

The following software/technology tools are required to run this project:

1. Python3: <https://www.python.org/downloads/>

2. Django framework: <https://www.djangoproject.com/download/>
3. MySQL workbench and its required tools: <https://dev.mysql.com/downloads/>

Steps to import the database are outlined below:

- a. The data files for mysql are found in the Dump folder. This is a top-level folder in the FACILEHIRE directory/folder.
- b. Create a database in MySQL workbench and name it “facile_hire”
- c. Click on the server menu, and then click on data import.
- d. Select the Dump folder from the FACILEHIRE folder and select “facile_hire” as the target schema.
- e. Begin the import
- f. Once done successfully, all tables and data should be present in the database.
4. Windows/linux/mac OS

1.4 E-R Model (see attached file)

2. Normalized Data Structure Tables, Relational Schema, and SQL Statements

2.1. Relational Schema

Below is a relational schema list after normalization.

1. **Address**(address_id(PK), street, zipcode(FK), apt_no)
2. **Background_verification**(background_verification_id(PK), status, created_date, modified_date)
3. **Benefits**(benefits_id(PK), benefit_name, benefit_charge, created_date, modified_date)
4. **Candidate**(candidate_id(PK), candidate_login_id(FK), name_id(FK), mobile, email, social_profiles, created_date, modified_date, skill, document_id(FK), education_id(FK), experience_id(FK), project_id(FK), college_id(FK), certification_id(FK), location_id(FK), background_verification_id(FK), company_id(FK), designation_id(FK), benefits(FK), status_id(FK), round_id(FK), schedule_id(FK))
5. **Candidate_login**(candidate_login_id(PK), user_name, password, created_date, modified_date)
6. **Certifications**(certifications_id(PK), certification_name, valid_from, valid_to, created_date, modified_date)
7. **Clients**(clients_id(PK), name_id(FK), created_date, modified_date)
8. **College**(college_id(PK), college_name, modified_date)
9. **Company**(company_id(PK), company_name, created_date,)
10. **Designation**(designation_id(PK), title, grade, created_date, modified_date, benefits_id(FK))
11. **Documents**(documents_id(PK), passport, experience_letters, education_documents, payslips, DL, ssn, created_date, modified_date, created_date)
12. **Education**(education_id(PK), degree, year_passed_out, created_date, modified_date)
13. **Employee**(employee_id(PK), employee_login_id(FK), name_id(FK), client_id((FK), mobile, created_date, modified_date, location_id(FK), candidate_id(FK), designation_id(FK), company_id(FK), status_id(FK))
14. **Employee_login**(employee_login_id(PK), user_name, password, created_date, modified_date)

- 15. Experience**(experience_id(PK), experience_name, from_date, to_date, created_date, modified_date)
- 16. Interview_call_status**(call_status_id (PK), contacted, not_answered, not_intrested, interested, remarks, created_date, modified_date)
- 17. Interview_round**(round_id(PK), interview round, remarks, created_date, modified_date)
- 18. Interview_Schedule**(interview_schedule_id(PK), schedule_interview, interview_scheduled, not_attended_interview, remarks, created_date, modified_date, interview_date)
- 19. Interview_status**(status_id(PK), selected, rejected, remarks, created_date, modified_date)
- 20. Job**(job_id (PK), job_name, requirements, created_date, modified_date, description, recruiter_id (FK), location_id(FK))
- 21. Location**(location_id(PK), address_id(FK), created_date, modified_date)
- 22. Name**(name_id(PK), first_name, last_name, middle_name)
- 23. Project**(project_id(PK), name, description, created_date, modified_date)
- 24. Recruiter**(recruiter_id(PK), mobile, created_date, modified_date, candidate_id(FK), document_id(FK), designation_id(FK), job_id(FK), location_id(FK), company_id(FK), call_status_id (FK), schedule_id (FK), round_id (FK))
- 25. Recruiter_login**(recruiter_login_id (PK). user_name, password, created_date, modified_date, recruiter_id (FK))
- 26. Resume**(resume_id(PK), name (FK), recruiter_id(FK), file_path, created_date, modified_date)
- 27. Zip**(zipcode(PK), city, state, country)

2.2 Relational Schema Diagram (see attached file)

2.3 SQL Statements

Include SQL statements to create tables based on the relational schema diagram you created.

1. Background_verfication

```
CREATE TABLE `background_verfication` (
  `background_verfication_id` int NOT NULL AUTO_INCREMENT,
  `status` varchar(350) DEFAULT NULL,
  `candidate_id` int DEFAULT NULL,
  `recruiter_id` int DEFAULT NULL,
  `employee_id` int DEFAULT NULL,
  `created_date` datetime DEFAULT CURRENT_TIMESTAMP,
  `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
```

```
PRIMARY KEY (`background_verfication_id`),  
UNIQUE KEY `candidate_id_UNIQUE` (`candidate_id`),  
UNIQUE KEY `recruiter_id_UNIQUE` (`recruiter_id`),  
UNIQUE KEY `employee_id_UNIQUE` (`employee_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

2.Candidate

```
CREATE TABLE `candidate` (  
    `candidate_id` int NOT NULL AUTO_INCREMENT,  
    `first_name` varchar(255) NOT NULL,  
    `middle_name` varchar(255) DEFAULT NULL,  
    `last_name` varchar(255) NOT NULL,  
    `email` varchar(255) NOT NULL,  
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
    `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
    `mobile` varchar(50) NOT NULL,  
    `skills` text,  
    PRIMARY KEY (`candidate_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=304 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Clients

```
CREATE TABLE `candidate` (  
    `candidate_id` int NOT NULL AUTO_INCREMENT,  
    `first_name` varchar(255) NOT NULL,  
    `middle_name` varchar(255) DEFAULT NULL,  
    `last_name` varchar(255) NOT NULL,  
    `email` varchar(255) NOT NULL,  
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,
```

```
`modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
`mobile` varchar(50) NOT NULL,  
`skills` text,  
PRIMARY KEY (`candidate_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=304 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

College

```
CREATE TABLE `candidate` (  
`candidate_id` int NOT NULL AUTO_INCREMENT,  
`first_name` varchar(255) NOT NULL,  
`middle_name` varchar(255) DEFAULT NULL,  
`last_name` varchar(255) NOT NULL,  
`email` varchar(255) NOT NULL,  
`created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
`modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
`mobile` varchar(50) NOT NULL,  
`skills` text,  
PRIMARY KEY (`candidate_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=304 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Company

```
CREATE TABLE `company` (  
`company_id` int NOT NULL AUTO_INCREMENT,  
`name` varchar(255) DEFAULT NULL,  
`created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
`modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
PRIMARY KEY (`company_id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=1883 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Designation

```
CREATE TABLE `designation` (  
    `designation_id` int NOT NULL,  
    `title` varchar(255) NOT NULL,  
    `grade` varchar(255) DEFAULT NULL,  
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
    `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
    PRIMARY KEY (`designation_id`,`title`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Documents

```
CREATE TABLE `documents` (  
    `document_id` int NOT NULL AUTO_INCREMENT,  
    `passport` varchar(255) DEFAULT NULL,  
    `experience_letters` varchar(255) DEFAULT NULL,  
    `education_documents` varchar(255) DEFAULT NULL,  
    `payslips` varchar(350) DEFAULT NULL,  
    `DL` varchar(255) DEFAULT NULL,  
    `ssn` varchar(45) DEFAULT NULL,  
    `candidate_id` int DEFAULT NULL,  
    `recruiter_id` int DEFAULT NULL,  
    `employee_id` int DEFAULT NULL,  
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
    `modified_date` datetime DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY (`document_id`),  
    UNIQUE KEY `candidate_id_UNIQUE` (`candidate_id`),
```

```
        UNIQUE KEY `recruiter_id_UNIQUE` (`recruiter_id`),
        UNIQUE KEY `employee_id_UNIQUE` (`employee_id`)
    ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
    COLLATE=utf8mb4_0900_ai_ci;
```

Education:

```
CREATE TABLE `education` (
    `education_id` int NOT NULL AUTO_INCREMENT,
    `degree` varchar(255) DEFAULT NULL,
    `year_passed_out` int DEFAULT NULL,
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,
    `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
    `college_id` int DEFAULT NULL,
    `cgpa` varchar(45) DEFAULT NULL,
    `course` varchar(100) DEFAULT NULL,
    `location_id` int DEFAULT NULL,
    PRIMARY KEY (`education_id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

Employee

```
CREATE TABLE `education` (
    `education_id` int NOT NULL AUTO_INCREMENT,
    `degree` varchar(255) DEFAULT NULL,
    `year_passed_out` int DEFAULT NULL,
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,
    `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
    `college_id` int DEFAULT NULL,
    `cgpa` varchar(45) DEFAULT NULL,
    `course` varchar(100) DEFAULT NULL,
```

```
`location_id` int DEFAULT NULL,  
PRIMARY KEY (`education_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Experience

```
CREATE TABLE `experience` (  
    `experience_id` int NOT NULL AUTO_INCREMENT,  
    `company_id` int DEFAULT NULL,  
    `from_date` varchar(45) DEFAULT NULL,  
    `to_date` varchar(45) DEFAULT NULL,  
    `designation_id` int DEFAULT NULL,  
    `description` text,  
    `location_id` int DEFAULT NULL,  
    PRIMARY KEY (`experience_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Interview_call_status

```
CREATE TABLE `interview_call_status` (  
    `interview_call_status_id` int NOT NULL AUTO_INCREMENT,  
    `contacted` varchar(45) DEFAULT NULL,  
    `not_answered` varchar(45) DEFAULT NULL,  
    `not_interested` varchar(45) DEFAULT NULL,  
    `remarks` text,  
    `candidate_id` int DEFAULT NULL,  
    `recruiter_id` int DEFAULT NULL,  
    `employee_id` int DEFAULT NULL,  
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
    `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
```

```
PRIMARY KEY (`interview_call_status_id`),  
UNIQUE KEY `candidate_id_UNIQUE` (`candidate_id`),  
UNIQUE KEY `recruiter_id_UNIQUE` (`recruiter_id`),  
UNIQUE KEY `employee_id_UNIQUE` (`employee_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Interview_round

```
CREATE TABLE `interview_call_status` (  
    `interview_call_status_id` int NOT NULL AUTO_INCREMENT,  
    `contacted` varchar(45) DEFAULT NULL,  
    `not_answered` varchar(45) DEFAULT NULL,  
    `not_interested` varchar(45) DEFAULT NULL,  
    `remarks` text,  
    `candidate_id` int DEFAULT NULL,  
    `recruiter_id` int DEFAULT NULL,  
    `employee_id` int DEFAULT NULL,  
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
    `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
PRIMARY KEY (`interview_call_status_id`),  
UNIQUE KEY `candidate_id_UNIQUE` (`candidate_id`),  
UNIQUE KEY `recruiter_id_UNIQUE` (`recruiter_id`),  
UNIQUE KEY `employee_id_UNIQUE` (`employee_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Interview_schedule

```
CREATE TABLE `interview_schedule` (  
    `interview_schedule_id` int NOT NULL AUTO_INCREMENT,  
    `interview_scheduled_status` varchar(45) DEFAULT NULL,
```

```

`interview_date` datetime DEFAULT NULL,
`remarks` text,
`candidate_id` int DEFAULT NULL,
`recruiter_id` int DEFAULT NULL,
`employee_id` int DEFAULT NULL,
`created_date` datetime DEFAULT CURRENT_TIMESTAMP,
`modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY (`interview_schedule_id`),
UNIQUE KEY `candidate_id_UNIQUE` (`candidate_id`),
UNIQUE KEY `recruiter_id_UNIQUE` (`recruiter_id`),
UNIQUE KEY `employee_id_UNIQUE` (`employee_id`)

) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

Interview_status

```

CREATE TABLE `interview_schedule` (
`interview_schedule_id` int NOT NULL AUTO_INCREMENT,
`interview_scheduled_status` varchar(45) DEFAULT NULL,
`interview_date` datetime DEFAULT NULL,
`remarks` text,
`candidate_id` int DEFAULT NULL,
`recruiter_id` int DEFAULT NULL,
`employee_id` int DEFAULT NULL,
`created_date` datetime DEFAULT CURRENT_TIMESTAMP,
`modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY (`interview_schedule_id`),
UNIQUE KEY `candidate_id_UNIQUE` (`candidate_id`),
UNIQUE KEY `recruiter_id_UNIQUE` (`recruiter_id`),

```

```
        UNIQUE KEY `employee_id_UNIQUE`(`employee_id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

Job

```
CREATE TABLE `interview_schedule` (
    `interview_schedule_id` int NOT NULL AUTO_INCREMENT,
    `interview_scheduled_status` varchar(45) DEFAULT NULL,
    `interview_date` datetime DEFAULT NULL,
    `remarks` text,
    `candidate_id` int DEFAULT NULL,
    `recruiter_id` int DEFAULT NULL,
    `employee` int DEFAULT NULL,
    `created_date` datetime DEFAULT CURRENT_TIMESTAMP,
    `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`interview_schedule_id`),
    UNIQUE KEY `candidate_id_UNIQUE`(`candidate_id`),
    UNIQUE KEY `recruiter_id_UNIQUE`(`recruiter_id`),
    UNIQUE KEY `employee_id_UNIQUE`(`employee_id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

Location

```
CREATE TABLE `location` (
    `location_id` int NOT NULL AUTO_INCREMENT,
    `street_address` varchar(255) DEFAULT NULL,
    `apt_no` varchar(255) DEFAULT NULL,
    `city` varchar(255) DEFAULT NULL,
    `state` varchar(350) DEFAULT NULL,
    `country` varchar(255) DEFAULT NULL,
```

```
`zipcode` int DEFAULT NULL,  
 `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
 `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
 PRIMARY KEY (`location_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=148041 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Recruiter

```
CREATE TABLE `location` (  
 `location_id` int NOT NULL AUTO_INCREMENT,  
 `street_address` varchar(255) DEFAULT NULL,  
 `apt_no` varchar(255) DEFAULT NULL,  
 `city` varchar(255) DEFAULT NULL,  
 `state` varchar(350) DEFAULT NULL,  
 `country` varchar(255) DEFAULT NULL,  
 `zipcode` int DEFAULT NULL,  
 `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
 `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
 PRIMARY KEY (`location_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=148041 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;
```

Resume

```
CREATE TABLE `resume` (  
 `resume_id` int NOT NULL AUTO_INCREMENT,  
 `name` varchar(255) DEFAULT NULL,  
 `file_path` varchar(255) DEFAULT NULL,  
 `created_date` datetime DEFAULT CURRENT_TIMESTAMP,  
 `modified_date` datetime DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,  
 `recruiter_id` int NOT NULL,
```

PRIMARY KEY (`resume_id`, `recruiter_id`)

) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

3. Sample Data and UI Component

3.1.1 Sample Data Loaded

Below are the tables that are loaded with sample data and used for our application.

Candidates table:

212	Michael	Donna	Robinson	michaelrobinson@gmail.com	2022-04-22 01:21:42	2022-04-22 15:32:17	5565924431	Video production,Indi
213	Cheryl	Jayden	Flores	cherylflores@yahoo.com	2022-04-22 01:21:42	2022-04-22 15:32:17	3682886267	Video production,Git,

Recruiter table:

11	Mary	6735364173	Recruiter	2022-04-17 14:25:57	2022-04-17 14:25:57	Jeffrey	Collins	marycollins@gmail.com
12	Ralph	9165522858	Senior Recruiter	2022-04-17 14:25:57	2022-04-17 14:25:57	Timothy	Miller	ralphmiller@rediff.com

Employee table:

```
1 •  SELECT * FROM facile_hire.employee;
```

Result Grid							
Edit: Export/Import: Wrap Cell Content:							
employee_id	first_name	middle_name	last_name	mobile	created_date	modified_date	email
1	Manohar	Kowthavarampu	123456789	2022-04-05 23:56:05	2022-04-24 21:05:35		kmanohar@gmail.com
2	James	Edwards	Miller	6675121634	2022-04-17 14:22:30	2022-04-17 14:22:30	jamesmiller@yahoo.com
3	Robert	Lincoln	Brown	6772943398	2022-04-17 14:22:30	2022-04-17 14:22:30	roberbrown@gmail.com
4	Elizabeth	Ruiz	Brown	1122192858	2022-04-17 14:22:30	2022-04-17 14:22:30	elizabethrizn@outlook.com
5	Linda	Elijah	Morales	7355971926	2022-04-17 14:22:30	2022-04-17 14:22:30	lindamorales@gmail.com
6	Mary	Richardson	Campbell	25326515499	2022-04-17 14:22:30	2022-04-17 14:22:30	marycampbell@outlook.com
7	Andrea	Amanda	Brown	4262311150	2022-04-17 14:22:30	2022-04-17 14:22:30	andreasbrown@yahoo.com
8	Aaron	Allen	Ward	3363475471	2022-04-17 14:22:30	2022-04-17 14:22:30	aaronward123@yahoo.com
9	Joseph	Morgan	Ward	2502153713	2022-04-17 14:22:30	2022-04-17 14:22:30	josephward123@yahoo.com
10	Kyle	Samuel	Wilson	9574824855	2022-04-17 14:22:30	2022-04-17 14:22:30	kylewilson@gmail.com
11	Ann	Joshua	Garcia	2685652979	2022-04-17 14:22:30	2022-04-17 14:22:30	annagarcia@outlook.com
12	James	Morgan	Brown	4427144642	2022-04-17 14:22:30	2022-04-17 14:22:30	jamesbrown@rediff.com

Certifications table:

12	Certified Secure Software Lifecycle Professional...	2022-04-25 03:33:39	HULL
----	---	---------------------	------

Clients table:

```
1 •  SELECT * FROM facile_hire.clients;
```

	clients_id	name	created_date	modified_date
▶	1	3scale	2022-04-20 13:42:25	NULL
	2	8tracks	2022-04-20 13:42:25	NULL
	3	10 by 10	2022-04-20 13:42:25	NULL
	4	15Five	2022-04-20 13:42:25	NULL
	5	21Tech	2022-04-20 13:42:25	NULL
	6	23andMe	2022-04-20 13:42:25	NULL
	7	80Twenty	2022-04-20 13:42:25	NULL
	8	99designs	2022-04-20 13:42:25	NULL
	9	140 Proof	2022-04-20 13:42:25	NULL
	10	2600Hz	2022-04-20 13:42:25	NULL
	11	Abl Schools	2022-04-20 13:42:25	NULL
	12	Abstract	2022-04-20 13:42:25	NULL

College table:

```
| 108 | Air Force Institute of Technology | 2022-04-20 12:53:49 | NULL |
```

Company table:

```
1 •  SELECT * FROM facile_hire.company;
```

	company_id	name	created_date	modified_date
	993	3scale	2022-04-20 13:42:25	NULL
	994	8tracks	2022-04-20 13:42:25	NULL
	995	10 by 10	2022-04-20 13:42:25	NULL
	996	15Five	2022-04-20 13:42:25	NULL
	997	21Tech	2022-04-20 13:42:25	NULL
	998	23andMe	2022-04-20 13:42:25	NULL
	999	80Twenty	2022-04-20 13:42:25	NULL
	1000	99designs	2022-04-20 13:42:25	NULL
	1001	140 Proof	2022-04-20 13:42:25	NULL
	1002	2600Hz	2022-04-20 13:42:25	NULL
	1003	Abl Schools	2022-04-20 13:42:25	NULL

Designation table:

```
| 7 | Principal Software Engineer | ? | 2022-04-23 14:50:43 | NULL |
```

Documents table:

```
| 1 | True | 205 | 1 | NULL | 2022-04-25 22:04:57 | 2022-04-2
```

Education table:

--	--	--	--	--	--	--	--	--	--

Experience table:

NULL									
------	------	------	------	------	------	------	------	------	------

Interview_call_status table:

NULL									
------	------	------	------	------	------	------	------	------	------

Interview_round:

▶ 1	True	1	NULL	205	True	True	True	Done with all rounds	True	2022
-----	------	---	------	-----	------	------	------	----------------------	------	------

Interview_schedule table:

▶ 1	interview_scheduled	2022-04-28 22:04:00	205	1	NULL	2022-04-25 20:39:19	2022-04-25 22:04:57
-----	---------------------	---------------------	-----	---	------	---------------------	---------------------

Interview_status table:

▶ 2	selected	205	1	NULL	2022-04-25 21:22:22	2022-04-25 22:04:57
-----	----------	-----	---	------	---------------------	---------------------

Job table:

▶ 17	Pythont Developer	Work experience as a Python developer • EXP...	2022-04-22 00:40:31	2022-04-22 01:03:07	We are looking for a Python deve...
15	Software Development Engineer AWS	Programming experience with at least one soft...	2022-04-25 03:53:05	2022-04-25 04:02:11	Amazon Web Services (AWS) is th...

Location table:

▶ 126808	Arab	Alabama	United States	2022-04-21 00:38:14	NULL
126808	NULL	NULL	NULL	2022-04-21 00:38:14	NULL

Resume table:

▶ 6	2_a_b_SG9nS0Y.pdf	/media/2_a_b_SG9nS0Y.pdf	2022-04-22 00:19:02	NULL	1
-----	-------------------	--------------------------	---------------------	------	---

Background_verification table:

▶ 1	initiated	205	1	NULL	2022-04-25 20:40:51	2022-04-25 22:04:57
-----	-----------	-----	---	------	---------------------	---------------------

Candidate_certifications table:

▶ 2	1	2
-----	---	---

Candidate_designation table:

▶ 11	214	26
------	-----	----

Candidate_education table:

*	NULL	NULL	NULL
---	------	------	------

Candidate_experience table:

▶ 3	1	3
-----	---	---

Candidate_job table:

▶ 6	205	14
-----	-----	----

Candidate_login:

▶ 1	candidate@g...	candidate123	2022-04-06 00:12:00	2022-04-24 20:47:25	1
-----	----------------	--------------	---------------------	---------------------	---

Employee_candidate table:

| 3 | 1 | 205

Employee_designation table:

| 10 | 10 | 15

Employee_login table:

| 5 | lindamorales@gmail.com | lindamorales123 | 2022-04-21 23:25:09 | NULL | 5

Recruiter_candidate table:

| 7 | 1 | 1

Recruiter_clients table:

| 1 | 1 | 1

Recruiter_company table:

| 1 | 1 | 1006

Recruiter_location table:

| 2 | — | 1 | 126801 | —

Recruiter_login:

```
1 •  SELECT * FROM facile_hire.recruiter_login;
```

result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	
recruiter_login_id	user_name	password	created_date	modified_date	recruiter_id
1	manohar@gmail.com	manohar123	2022-04-05 23:48:16	2022-04-21 23:12:39	1
102	stephengarcia@rediff.com	stephengarcia123	2022-04-21 23:23:14	NULL	2
103	pamelahughes@outlook.com	pamelahughes123	2022-04-21 23:23:14	NULL	3
104	cherylcollins@outlook.com	cherylcollins123	2022-04-21 23:23:14	NULL	4
105	michaelmorris@rediff.com	michaelmorris123	2022-04-21 23:23:14	NULL	5
106	patriciarobinson@rediff.com	patriciarobinson123	2022-04-21 23:23:14	NULL	6
107	johnrobinson@gmail.com	johnrobinson123	2022-04-21 23:23:14	NULL	7
108	annbrown@rediff.com	annbrown123	2022-04-21 23:23:14	NULL	8
109	stephenmorales@outlook.com	stephenmorales123	2022-04-21 23:23:14	NULL	9
110	kylecarter@outlook.com	kylecarter123	2022-04-21 23:23:14	NULL	10

3.1.2 Functions and Queries

get_mysql_connection method is used to get the mysql connection and cursor objects

execute_query_fetchone method is used to fetch the one record from the mysql for the given query

| RETURN RESULTS

Execute_query_fetchall method is used to fetch the multiple records from the database for the given query

```
| return results
```

execute_query method is used to execute the given query and commit to database

```
|
```

Candidate_login method is used to check the whether candidate is valid or not

```
| return context, template
```

Candidate_dashboard method is used to fetch the candidates jobs details that has been applied, recruiter contact details.

```
| return render(request, "hire/candidate_dashboard.html", context)
```

Employee_login method is used to validate the employee is valid or not.

```
| return login_status, template
```

Employee_dashboard method is used to fetch the candidate that are assigned to the employee for the interviews.

```
| return render(request, "hire/employee_dashboard.html", context)
```

Employee_candidate method is used to fetch the candidate details when employee clicks on a particular candidate_id

```
| return render(request, "hire/employee_candidate.html", context=context)
```

Recruiter_login method is used to check whether the recruiter is valid or not.

```
| return login_status, template
```

Recruiter_dashboard method is used to fetch the details of that particular recruiter that, how many resumes uploaded, how many jobs posted, how many candidates are there.

```
| return render(request, "hire/recruiter-dashboard.html", context)
```

Recruiter_jobs method is used to fetch the jobs for respective recruiter and their details.

```
| return render(request, "hire/recruiter-jobs.html", context=context) You, yesterday • more changes
```

Recruiter_candidates method is used to fetch the candidates to the respective recruiter

```
| return render(request, "hire/recruiter-candidates.html", context)
```

Recutier_resumes method is used to upload the resumes to local path and insert the details of the resumes in database

```
| return render(request, "hire/recruiter-resumes.html", context)
```

Recruiter_profile method is used to fetch the recruiter details

```
| context["age"] = random.randint(25, 40)
```

Populate_call_status method is used to populate the call status details of candidate when recruiter contacted the candidate.

```
|
```

Populate_interview_schedule method is used to populate the interview schedule details when recruiter updated to the respective candidate

```
)
```

Populate_interview_round method is used to populate the interview details of the candidate

```
technical_evaluation, mangerial_round, hr_round, final_round, interview_round_remarks))
```

Populate_background_verification method is used to populate the background verification details once the candidate is selected

```
verification))
```

Populate_interview_status method is used to populate the interview status of the candidate whether he is selected or rejected.

```
(recruiter_id), interview_status, interview_status_remarks))
```

Populate_job_details method is used to populate the job details for that respective candidate

```
)
```

Populate_documents method is used to populate the documents status that candidate has been submitted.

```
candidate_id, localStorage.getItem("recruiter_id"), passport, experience_letter,  
educational_documents, payslips, dl, ssn))
```

Recruiter_candidate_profile method is used to fetch the candidate profile as well to update the candidate interview details.

```
documents = get_documents(candidate_id, localStorage.getItem("recruiter_id"))
```

Post_job method is used to post the job by the recruiter.

```
context['message'] = message  
return render(request, "hire/post_job.html", context=context)
```

Get_location_id method is used to fetch the location id.

```
return result[0]
```

Recruiter_detail_job method is used to fetch the full details for the respective job selected by recruiter

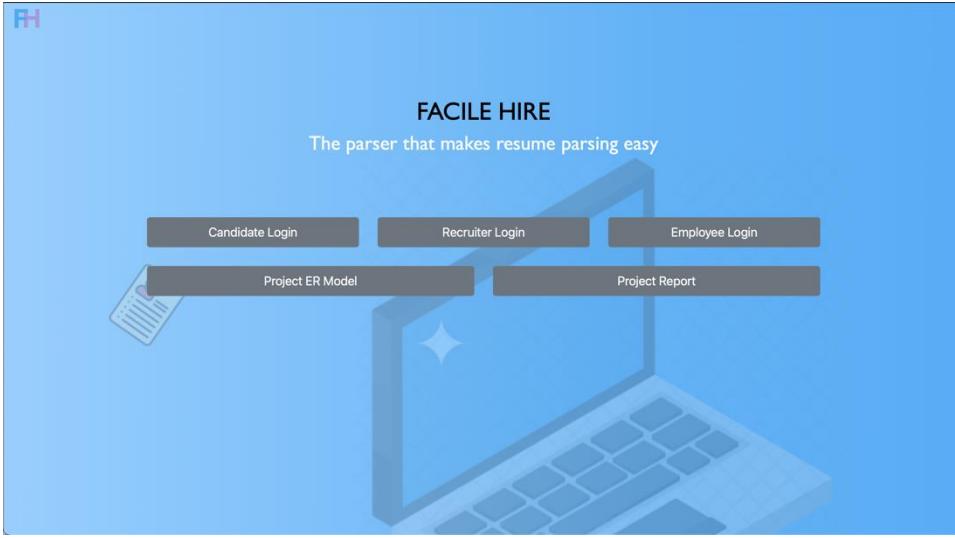
```
return render(request, "hire/recruiter_detail_job.html", context=context)
```

Candidate_profile_view method is used to build the candidate profile.

```
})
```

3.2 UI Component

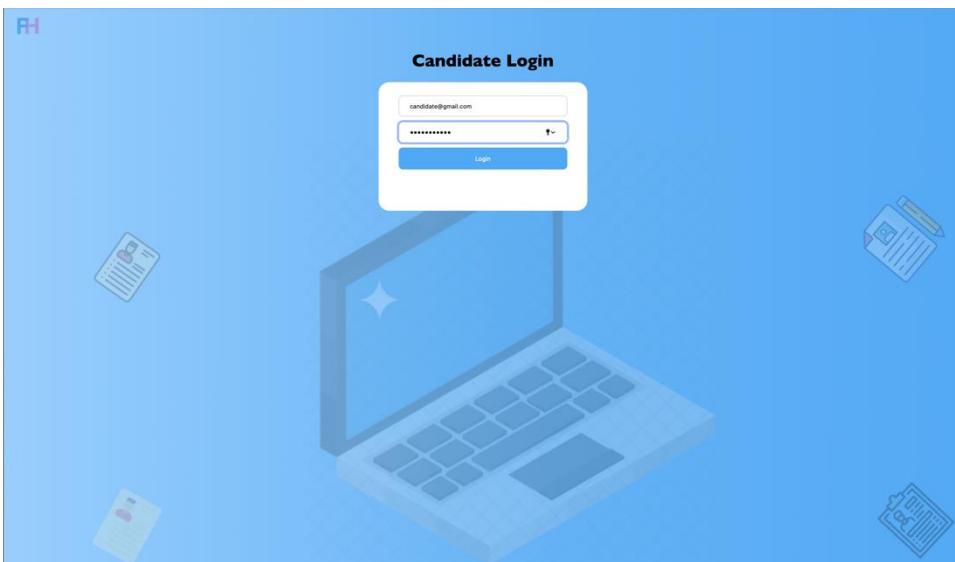
3.2.1 Homepage

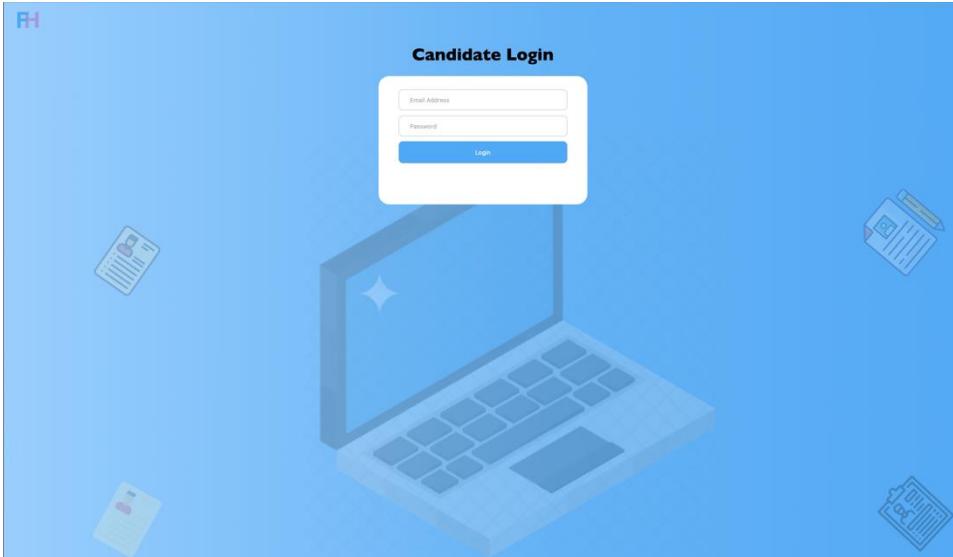


3.2.2 Candidate Portal

3.2.2.1 Logging into the candidate portal

The Candidate can log in using login credentials and clicking login button.





3.2.2.2 Viewing the Candidate's dashboard

In the Candidate dashboard, the candidate can view the status of the job they have applied to.

 A screenshot of a web-based candidate dashboard. At the top, there are navigation links: "JOB STATUS" (highlighted in blue), "PROFILE", and "LOGOUT". Below the header, the job title is listed as "Python Developer". The job description is: "We are looking for a Python Developer to join our engineering team and help us develop and maintain various software products. Python Developer responsibilities include writing and testing code, debugging programs and integrating applications with third-party web services. To be successful in this role, you should have experience using server-side logic and work well in a team. Ultimately, you'll build highly responsive web applications that align with our business needs." The job requirements are: "Work experience as a Python Developer - Expertise in at least one popular Python framework (like Django, Flask or Pyramid) - Knowledge of object-relational mapping (ORM) - Familiarity with front-end technologies (like JavaScript and HTML5) - Team spirit - Good problem-solving skills". The job responsibilities are: "Write effective, scalable code - Develop back-end components to improve responsiveness and overall performance - Integrate user-facing elements into applications - Test and debug programs - Improve functionality of existing systems - Implement security and data protection solutions - Assess and prioritize feature requests - Coordinate with internal teams to understand user requirements and provide technical solutions". The job location is "Alabama United States". Recruiter details are: "Name: Manohar", "Email: manohar@gmail.com", and "Mobile: 123456789". Under "Document(s) Submitted", there are checkboxes for "Passport", "Experience Letter(s)", "Educational Document(s)", and "Paylip(s)". Under "Cell Status", there are radio buttons for "Contacted", "Not Answered", "Not Interested", and "Interested". Under "Interview Schedule", there are checkboxes for "Schedule Interview", "Interview Scheduled", and "Not attended Interview". The interview date is "04/24/2022, 12:30 PM". Under "Background Verification", there are radio buttons for "Initiated", "In Progress", and "Completed". Under "Interview Round", there are checkboxes for "Written Test", "Technical Evaluation", "Mangerial Round", and "HR Round", with "Final Round" also listed. Under "Interview Status", there are radio buttons for "Selected" and "Rejected".

3.2.2.3 Viewing the candidate's profile/resume

The candidates' profile is displayed in the format of a standard resume and can be accessed when the candidate clicks the profile button after logging into their portal. Each candidate's profile/resume would contain details which are used to match the specific candidate to jobs for which they qualify.

JONATHAN DOE
WEB DESIGNER, DIRECTOR

Profile
name@yourdomain.com
(555) 867-2345

Skills
Web Design
Assertively exploit wireless initiatives rather than synergistic core competencies.

Interface Design
Credibly streamline mission-critical value with multifunctional functionalities.

Project Direction
Proven ability to lead and manage a wide variety of projects in team and independent situations.

Technical
XHTML
CSS
Javascript

Jquery
PHP
CVS / Subversion

OS X
Windows XP/Vista
Linux

Experience
Facebook Senior Interface Designer 2005-2007
Intricately enable optimal core competencies through corporate relationships. Proactively implement worldwide vertical and client-focused imperatives. Conveniently initiate virtual paradigms and top-line convergence.

Apple Inc.
Senior Interface Designer 2005-2007
Progressively reorchestrate multifunctional "outside the box" thinking through transparent methods of empowerment. Compellingly morph extensive niches marked with mission-critical ideas. Proactively deliver bricks-and-clicks strategic theme areas rather than isolable

JOBSITE | PROFILE | LOGOUT

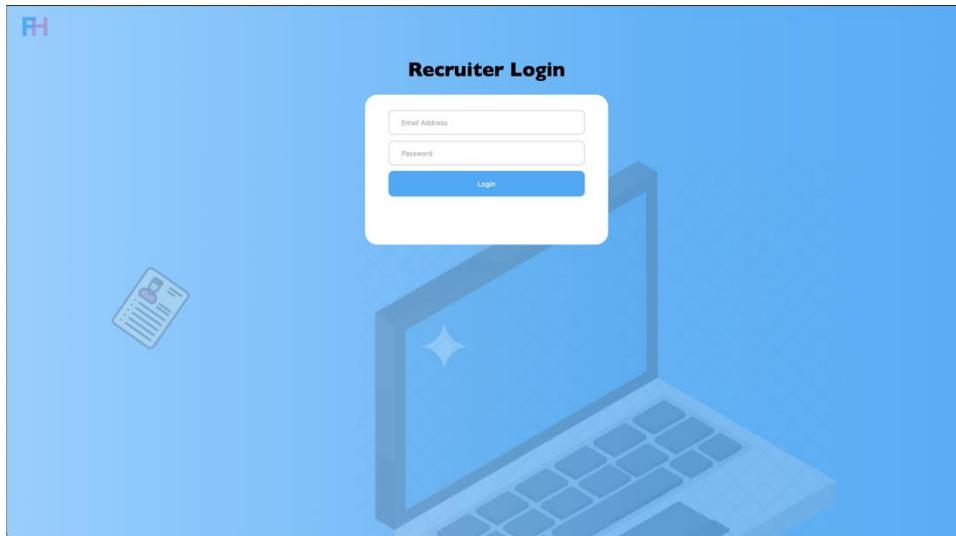
3.2.2.4 Logging out of the candidate portal

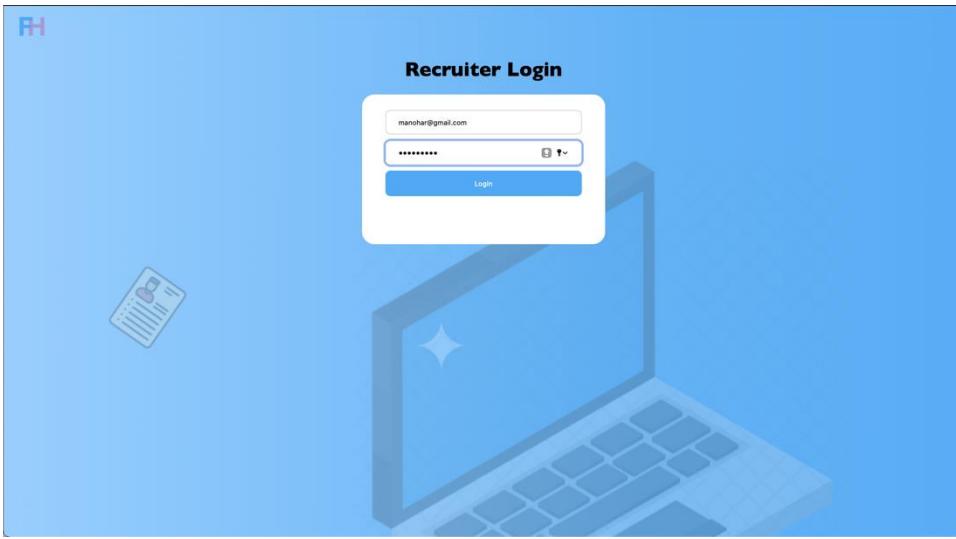
The candidate can successfully log out of their portal by clicking the logout button located at the top right corner of the page.

3.2.3 Recruiter Portal

3.2.3.1 Logging into the recruiter portal

The Recruiter can log in using login credentials and clicking login button.





3.2.3.2 Viewing the recruiter Dashboard

In the recruiter dashboard, the recruiter can have access to all the candidates, scheduled interview details, Candidates who were hired, all the resumes uploaded, and open positions.



3.2.3.3 Viewing the recruiter Jobs page

The screenshot shows a table with one row of data. The columns are Job ID, Position, Status, Date Created, and Date Modified. The data row contains: Job ID 14, Position Python Developer, Status open, Date Created April 22, 2022, 12:48 a.m., and Date Modified April 22, 2022, 1:05 a.m. A 'DETAILS' button is located at the bottom right of the table row.

The jobs page located within the recruiter portal contains the list of jobs which have been added and assigned to or added by a specific recruiter. Each job record displays the job ID, position, roles, date created, and date modified, for a specified job posting. Upon clicking the details button for a specific job record, a new page opens, displaying all details of that particular job posting (figure below).

This screenshot shows a detailed view of a job posting. The job title is 'Python Developer'. The job description states: 'We are looking for a Python Developer to join our engineering team and help us develop and maintain various software products. Python Developer responsibilities include writing and testing code, debugging programs and integrating applications with third-party web services. To be successful in this role, you should have experience using server-side logic and work well in a team. Ultimately, you'll build highly responsive web applications that align with our business needs.' The job requirements mention expertise in Python frameworks like Django, Flask or Pyramid, knowledge of object-relational mapping (ORM), and familiarity with front-end technologies like JavaScript and HTML5. The job responsibilities include writing effective, scalable code, developing back-end components, integrating user-facing elements, testing and debugging programs, improving functionality, implementing security, and prioritizing feature requests. The job location is Alabama United States. The recruiter's details are listed as Name: Manohar, Email: manohar@gmail.com, and Mobile: 123456789. Document submission options include Passport, Experience Letter(s), Educational Document(s), and Paylip(s). The cell status section includes radio buttons for Contacted, Not Answered, Not Interested, and Interested. Interview schedule options include Schedule Interview, Interview Scheduled, and Not attended Interview. Interview round options include Written Test, Technical Evaluation, Managerial Round, and HR Round. Final round is also listed. Interview date is 04/24/2022, 12:30 PM. Background verification status includes Initiated, In Progress, and Completed. Interview status includes Selected and Rejected.

When done, one can return to the previous page to view other job postings.

3.2.3.4 Uploading Resumes to Recruiter Portal

The recruiter has the ability to upload candidate resumes to the portal. To upload a resume, the recruiter must click on the “upload resume” button, after which they will be redirected to the page showing in the figure below.

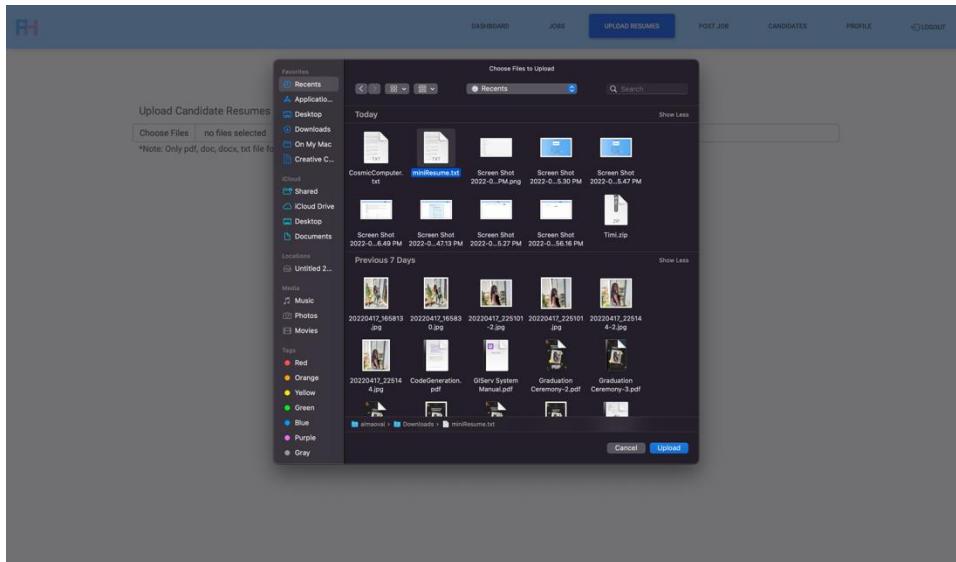


Upload Candidate Resumes

Choose Files no files selected

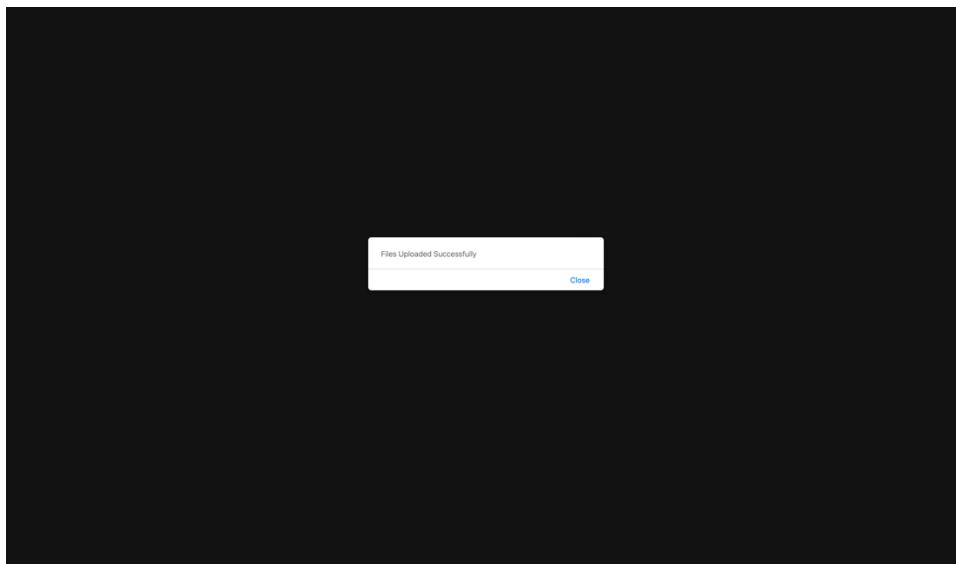
*Note: Only pdf, doc, docx, txt file formats are supported

The recruiter can then select what document they wish to upload from their local system and click on the upload button.





Upon successful upload of the resume, a feedback message will be displayed to the recruiter to let them know that the resume has been uploaded successfully as shown in the figure below.



3.2.3.5 Uploading Jobs on Recruiter Portal

In order for a recruiter to post a new job, they would first need to visit the “post job” page. Then, they will be redirected to a webpage which has different input boxes for the different job details. All fields require regular text entry, except the job location and job status fields, which can be selected from dropdown menus.

The screenshot shows a 'Post Job' form on a recruitment platform. The top navigation bar includes links for DASHBOARD, JOBS, UPLOAD RESUMES, POST JOB (which is highlighted in blue), CANDIDATES, PROFILE, and LOGOUT. The main form has sections for Job Title, Job Location, and Job Status. Below these are fields for Requirements, Roles, and Job Description, each with a text input area. A purple 'SUBMIT' button is located at the bottom right of the form.

Job Title: Title

Job Location: Alabama, United States

Job Status: open

Requirements:

Roles:

Job Description:

Post Job

SUBMIT

This screenshot shows the same 'Post Job' form after the user has entered data. The Job Title is 'Lead Engineer II', the Job Location is 'Texas, United States', and the Job Status is 'open'. The Requirements field contains '2 year Associates degree with 4 years experience. 4 year Bachelor preferred, with 1 year experience.' The Roles field contains 'perform code reviews' and 'Lead programmer teams on development projects.'. The Job Description field contains the text: 'T is looking for a passionate individual to take up the role of Lead Engineer II in the Texas team. This position comes with benefits and is a hybrid position in the city of Austin, Texas.' A purple 'SUBMIT' button is visible at the bottom right.

Job Title: Lead Engineer II

Job Location: Texas, United States

Job Status: open

Requirements:

2 year Associates degree with 4 years experience.
4 year Bachelor preferred, with 1 year experience.

Roles:

perform code reviews
Lead programmer teams on development projects.

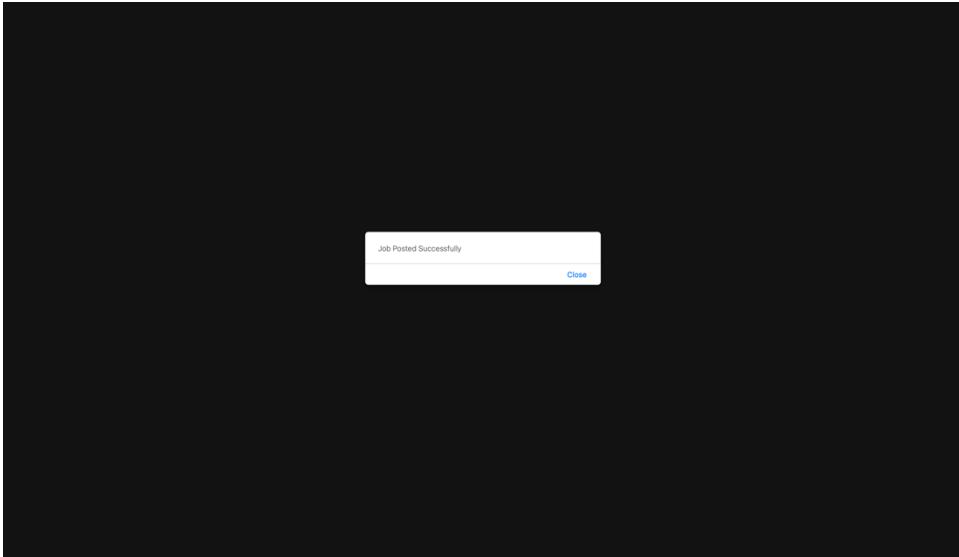
Job Description:

T is looking for a passionate individual to take up the role of Lead Engineer II in the Texas team. This position comes with benefits and is a hybrid position in the city of Austin, Texas.

Post Job

SUBMIT

When the recruiter has entered the details of the job they desire to post, they can click on the submit button, and upon successful posting, they will be presented with a successful feedback message such as the one in the figure below.



Job Posted Successfully

[Close](#)

The newly posted job can now be seen on the jobs list and the details can also be seen as described in prior section and shown in the figures below.

A screenshot of a web application interface titled 'JOBS'. The page displays a table of job listings with columns: Job ID, Position, Status, Date Created, and Date Modified. Two rows are visible:

Job ID	Position	Status	Date Created	Date Modified	Action
14	Python Developer	open	April 22, 2022, 12:48 a.m.	April 22, 2022, 1:05 a.m.	DETAILS
15	Lead Engineer II	None	April 25, 2022, 12:03 a.m.	None	DETAILS

Job ID:
15

Job Created:
April 25, 2022, 12:03 a.m.

Job Title:
Lead Engineer II

Job Description:
T1 is looking for a passionate individual to take up the role of Lead Engineer II in the Texas team. This position comes with benefits and is a hybrid position in the city of Austin, Texas.

Job Requirements:
2 year Associates degree with 4 years experience. 4 year Bachelor preferred, with 1 year experience.

Job Responsibilities:
Perform code reviews Lead programmer teams on development projects.

Job Location:
Texas, United States

3.2.3.6 Viewing Candidates from Recruiter Portal

The recruiter can see a list of all candidates that have been assigned or matched to him/her through the jobs they have been assigned. To view the profile of any specific candidate, the recruiter can click on the “VIEW PROFILE” link located to the far right of the candidate they wish to view. An example is shown in the figure below.

Name	Title	Position	Mobile	Skills	Status	Actions
Jennifer Logan Martin jennifermartin@gmail.com	Software engineer IT department	Senior	3596782716	Azure,Industrial design,Data science,Ruby,C/C++,Javascript,Git,AWS,Docker,Video production	Active	VIEW PROFILE
Joseph Jayden Brown josephbrown@gmail.com	Software engineer IT department	Senior	8363614526	Git,Data science,C/C++,Industrial design,Docker,Azure,Video production,Ruby,AWS,JavaScript	Active	VIEW PROFILE
Kyle Morgan Walker kylewalker@rediff.com	Software engineer IT department	Senior	62971196424	Video production,JavaScript,Git,Ruby,Docker,AWS,C/C++,Azure,Industrial design,Data science	Active	VIEW PROFILE
Linda Edwards Rivera lindarivera@rediff.com	Software engineer IT department	Senior	7261469928	Docker,Azure,Video production,Git,JavaScript,Ruby,AWS,C/C++,Industrial design,Data science	Active	VIEW PROFILE
Mary Richardson Miller marymiller@gmail.com	Software engineer IT department	Senior	4822657974	JavaScript,AWS,Ruby,Docker,Data science,Azure,Git,Video production,C/C++,Industrial design	Active	VIEW PROFILE
Michael Cynthia Garcia michaelgarcia@outlook.com	Software engineer IT department	Senior	9914535811	Git,Azure,Industrial design,Video production,Ruby,JavaScript,Docker,C/C++,AWS,Data science	Active	VIEW PROFILE

Show 10 entries

Search:

Previous | 1 | Next

Upon clicking to view the candidates’ profile, the recruiter will be redirected to a page through which they can view the basic details about the candidate, as well as the current status of their profile, regarding any jobs they have been matched to, or any interview or employment process they are going through or have completed.

Candidate Profile Status

Name Joseph Jayden Brown	Email josephbrown@gmail.com	Position Senior Software Engineer	Mobile 8363614526
Skills Git,Data science,C/C++,Industrial design,Docker,Azure,Video production,Ruby,AWS,Javascript		Click here to view candidate profile	
Document(s) <input type="checkbox"/> Passport <input type="checkbox"/> Experience Letter(s) <input type="checkbox"/> Educational Document(s) <input type="checkbox"/> Payslip(s)		Call Status <input type="radio"/> Contacted <input type="radio"/> Not Answered <input type="radio"/> Not Interested <input type="radio"/> Interested	
Interview Schedule <input type="radio"/> Schedule Interview <input type="radio"/> Interview Scheduled <input type="radio"/> Not attended Interview Interview Date 04/25/2022, 12:30 PM		Interview Round <input type="checkbox"/> Written Test <input type="checkbox"/> Technical Evaluation <input type="checkbox"/> Managerial Round <input type="checkbox"/> HR Round <input type="checkbox"/> Final Round	
Remarks <input type="text"/>		Remarks <input type="text"/>	
Background Verification <input type="radio"/> Initiated <input type="radio"/> In Progress <input type="radio"/> Completed		Interview Status <input type="radio"/> Selected <input type="radio"/> Rejected	
Job ID <input type="text"/>		Remarks <input type="text"/>	
SUBMIT			

These details on the candidates' profile status page can also be edited or updated by the recruiter accordingly, and the information would also be updated on the job status page of the specified candidate.

Candidate Profile Status

Name Joseph Jayden Brown	Email josephbrown@gmail.com	Position Senior Software Engineer	Mobile 8363614526
Skills Git,Data science,C/C++,Industrial design,Docker,Azure,Video production,Ruby,AWS,Javascript		Click here to view candidate profile	
Document(s) <input checked="" type="checkbox"/> Passport <input checked="" type="checkbox"/> Experience Letter(s) <input checked="" type="checkbox"/> Educational Document(s) <input type="checkbox"/> Payslip(s)		Call Status <input type="radio"/> Contacted <input type="radio"/> Not Answered <input type="radio"/> Not Interested <input type="radio"/> Interested	
Interview Schedule <input type="radio"/> Schedule Interview <input type="radio"/> Interview Scheduled <input type="radio"/> Not attended Interview Interview Date 04/25/2022, 12:30 PM		Interview Round <input type="checkbox"/> Written Test <input type="checkbox"/> Technical Evaluation <input type="checkbox"/> Managerial Round <input type="checkbox"/> HR Round <input type="checkbox"/> Final Round	
Remarks <input type="text"/>		Remarks <input type="text"/>	
Background Verification <input checked="" type="radio"/> Initiated <input type="radio"/> In Progress <input type="radio"/> Completed		Interview Status <input type="radio"/> Selected <input type="radio"/> Rejected	
Job ID <input type="text" value="14"/>		Remarks <input type="text"/>	
SUBMIT			

3.2.3.7 Viewing the Recruiter's Profile

Like the candidates, recruiters also have unique profile pages with all their basic contact information.

Manohar
Recruiter in Accenture

Client	3scale	E-mail	manohar@gmail.com
Age	28 Yr	Phone	123456789
Location	Alabama, United States	Skype	manohar

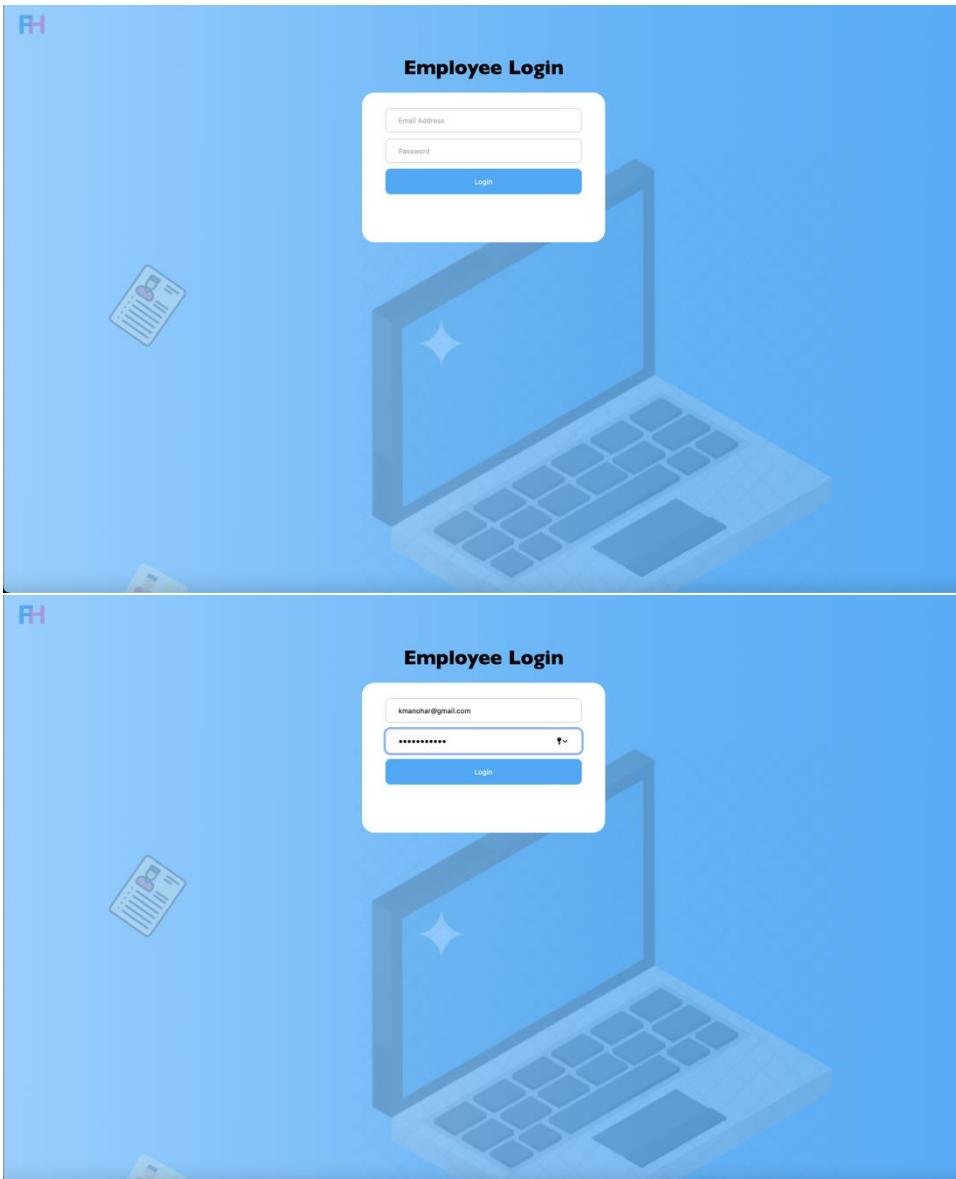
3.2.3.8 Logging out of recruiter profile

The recruiter can log out of their profile successfully by clicking the logout button at the top right corner of the webpage.

3.2.4 Employee Portal

3.2.4.1 Logging into the Employee Portal

The Employees can log in using login credentials and clicking login button.



3.2.4.2 The Employee Portal Dashboard

In the employees Dashboard, employees can view details of the employees

Candidates						
Candidate List						Logout
Name		Title	Position	Mobile	Skills	
 candidate new	candidate@gmail.com	Software engineer	Senior	123456789	Javascript,Industrial design,Docker,Ruby,Git,Video production,Data science,AWS,Azure,C/C++	Active View Profile
 Jennifer Logan Martin	jennifermartin@gmail.com	Software engineer	Senior	3596782716	Azure,Industrial design,Data science,Ruby,C/C++,Javascript,Git,AWS,Docker,Video production	Active View Profile
 Mary Richardson Miller	marymiller@gmail.com	Software engineer	Senior	4822657974	Javascript,AWS,Ruby,Docker,Data science,Azure,Git,Video production,C/C++,Industrial design	Active View Profile

3.2.4.3 Viewing a candidate's profile status from employee portal

Candidates						
Candidate List						Logout
Name		Title	Position	Mobile	Skills	
 candidate new	candidate@gmail.com	Software engineer	Senior	123456789	Javascript,Industrial design,Docker,Ruby,Git,Video production,Data science,AWS,Azure,C/C++	Active View Profile
 Jennifer Logan Martin	jennifermartin@gmail.com	Software engineer	Senior	3596782716	Azure,Industrial design,Data science,Ruby,C/C++,Javascript,Git,AWS,Docker,Video production	Active View Profile
 Mary Richardson Miller	marymiller@gmail.com	Software engineer	Senior	4822657974	Javascript,AWS,Ruby,Docker,Data science,Azure,Git,Video production,C/C++,Industrial design	Active View Profile

Candidate Profile Status

Name: Jennifer Logan Martin Email: jennifermartin@gmail.com Position: Big Data Architect Mobile: 35967822716

Skills: Azure,Industrial design,Data science,Ruby,C/C++,JavaScript,Git,AWS,Docker,Video production [Click here to view candidate profile](#)

Interview Schedule
 Schedule Interview Interview Scheduled Not attended Interview
 Interview Date: 04/26/2022, 12:40 AM

Interview Round
 Written Test Technical Evaluation Managerial Round HR Round
 Final Round

Remarks: Interview with TI Software team.

Interview Status
 Selected Rejected
 Remarks:

SUBMIT

3.2.4.4 Logging out of employee portal

The Facile Hire employee can log out of their portal by simply clicking the logout button, and they would be redirected to the home page.

4. Appendix

GitHub Repository: <https://github.com/aimaovai/FacileHire.git> or <https://github.com/mkowthavarapu/FacileHire>

5. References

N/A