

A report on

# **SURVEILLANCE AGENCY SYSTEM**

By

2018103033 | KAPILESH GURUPRASAD

2018103518 | ASWIN R R V

TEAM NUMBER 16

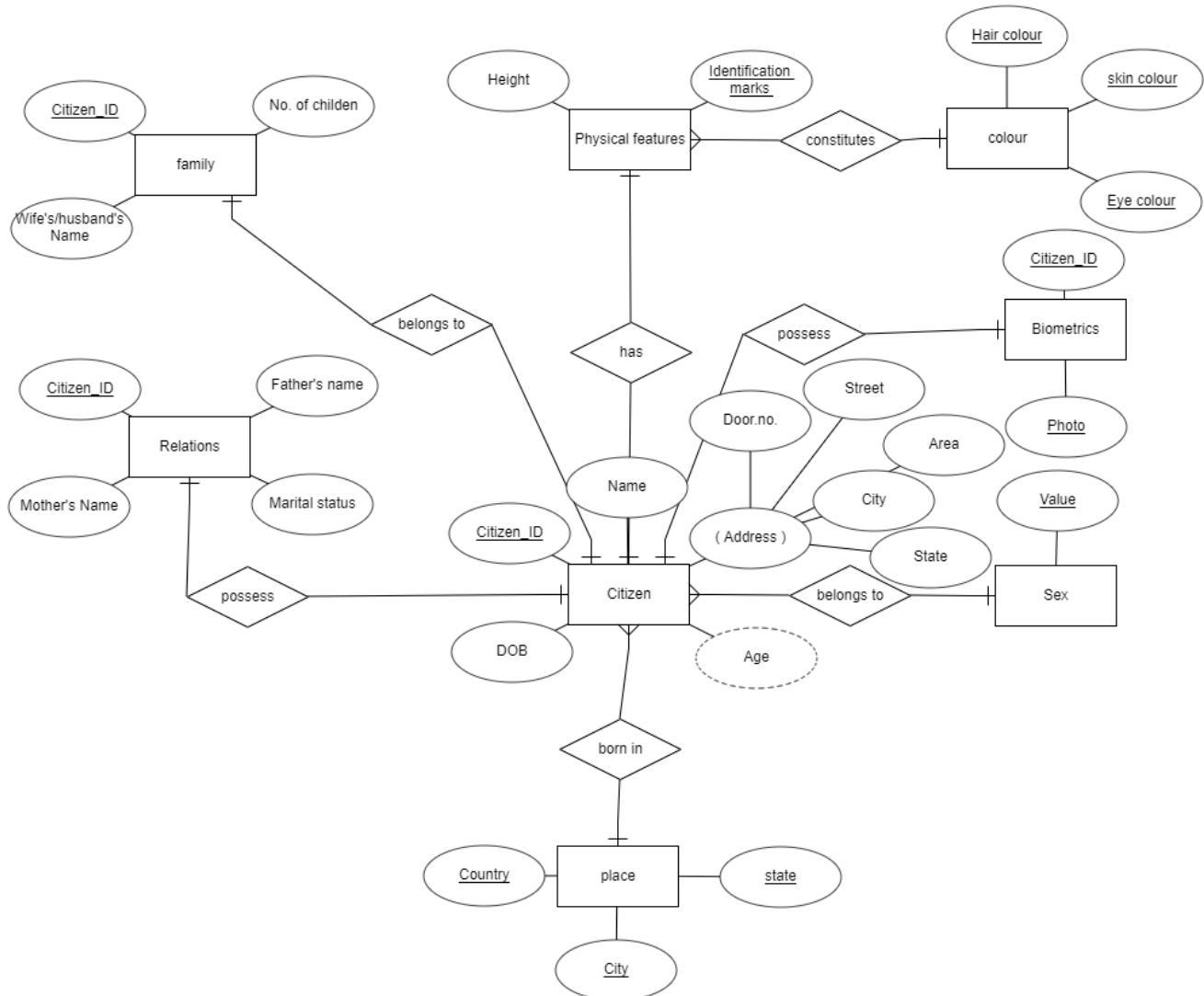
Submitted for the course

CS7411- Database Management Systems Laboratory

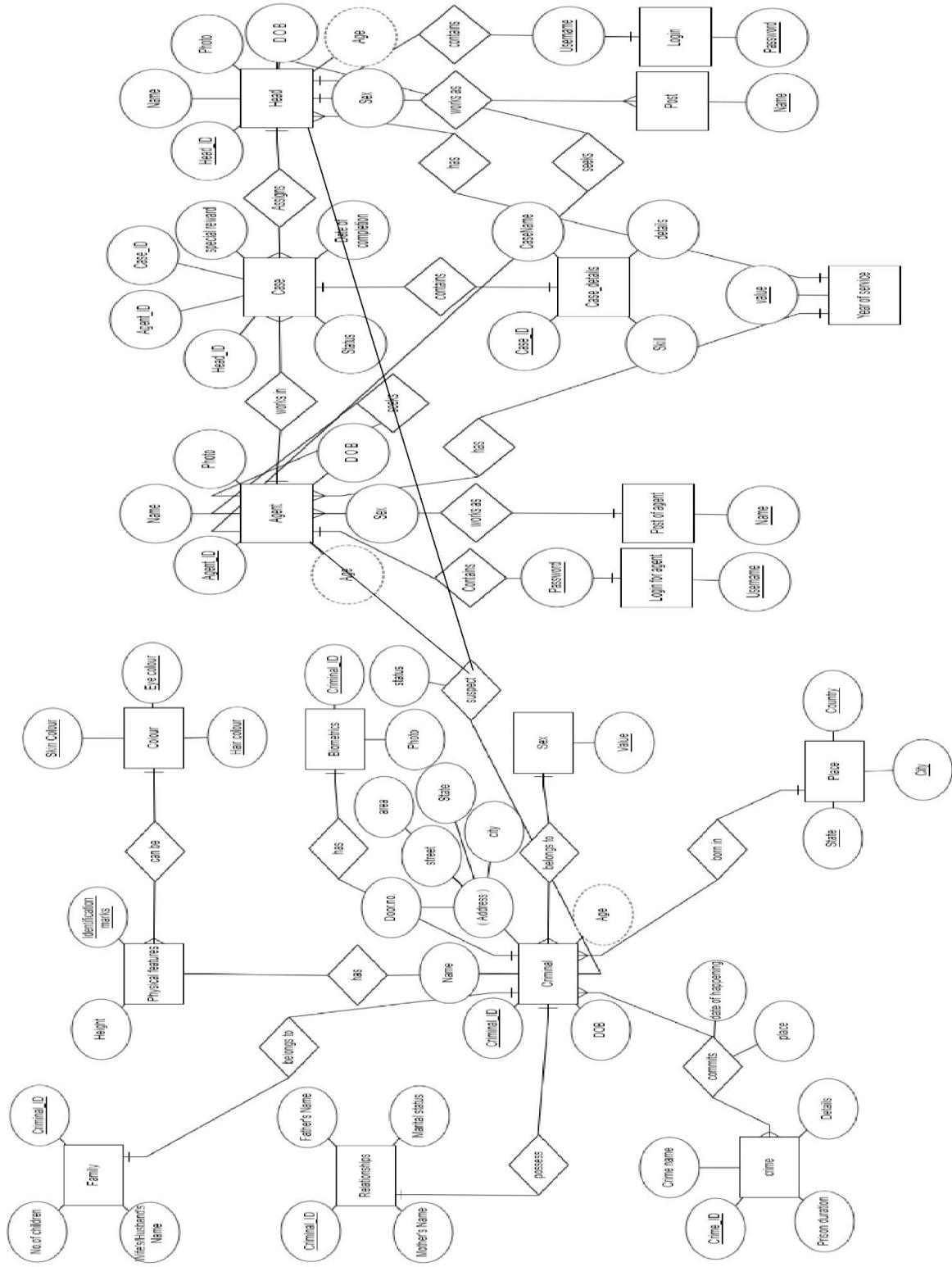
	Max	Assigned	
SQL	7	10	
Views	3	3	
Procedures	5	3	
Functions	5	4	Evaluator Name and Signature
Triggers	5	4	Date: 10 – 04 – 2020
Total		24	

# ER Diagram

## DATABASE-1:-PUBLIC DATABASE

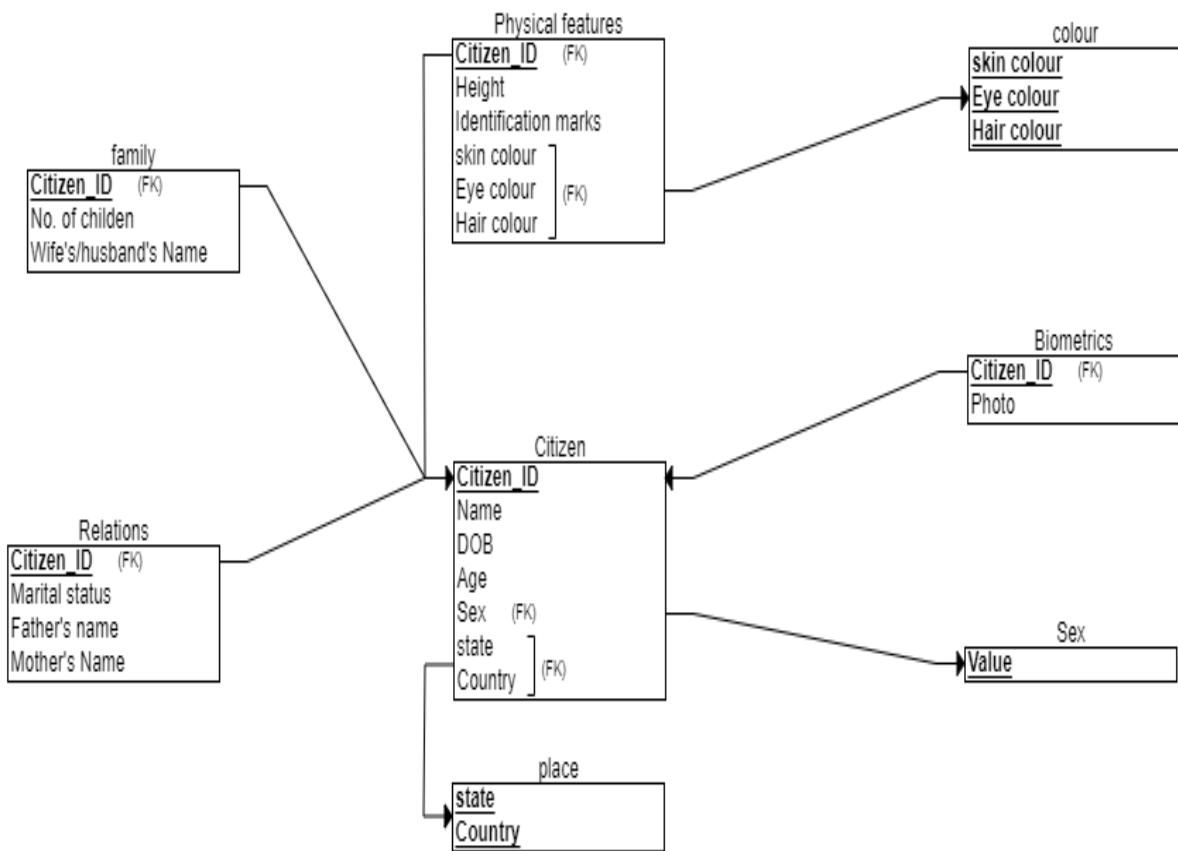


# DATABASE 2:- OFFICIAL DATA BASE

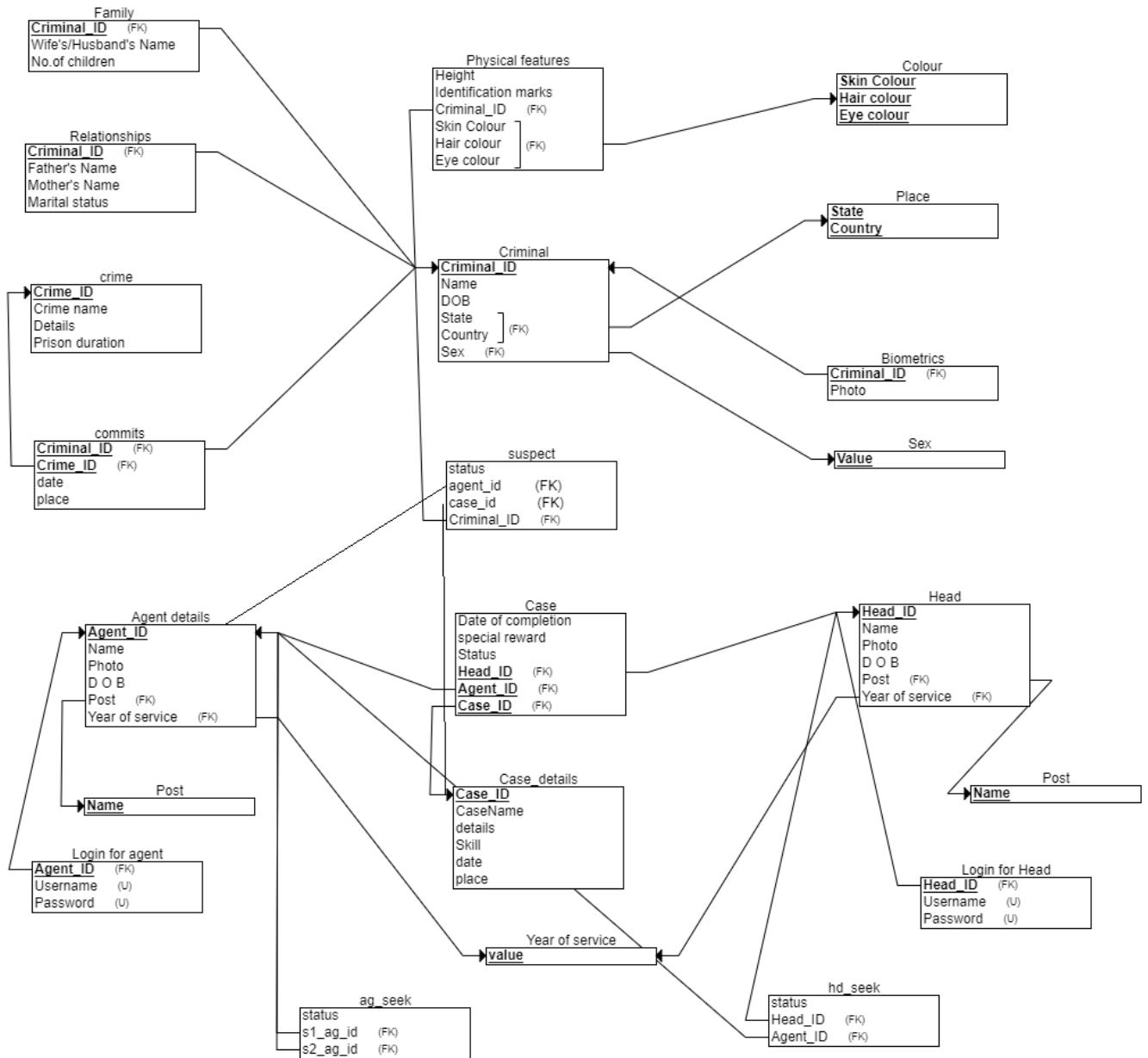


# DB Schema

## DATABASE-1:-PUBLIC DATABASE



## DATABASE 2:- OFFICIAL DATA BASE



# Instances of each Relation (Snapshot)

## DATABASE-1:-PUBLIC DATABASE

### 1.CITIZEN

The screenshot shows the pgAdmin 4 interface with the 'Public' schema selected. The 'citizen' table is displayed in a data grid. The columns are: citizen\_id, name, dob, age, sex, state, and country. The data consists of 15 rows of citizen information.

citizen_id	name	dob	age	sex	state	country
1	501 Dhanush	2000-01-01	20	M	TN	India
2	503 Vijay	1978-06-01	41	M	TN	India
3	504 Sethu	1983-10-01	36	M	TN	India
4	505 Donald	1952-08-01	67	M	MH	India
5	506 Mallya	1959-07-01	60	M	UP	India
6	507 Nirav	1966-04-01	53	M	UP	India
7	508 Pawan	1975-02-01	44	M	AP	India
8	509 Chandrababu	1951-03-01	68	M	AP	India
9	510 kala	1969-09-01	50	F	TN	India
10	511 Arjun	1989-08-01	30	M	AP	India
11	512 Karthikeyan	1989-11-01	30	M	TN	India
12	513 Andrea	1990-12-01	29	F	MP	India
13	514 Pandey	1986-05-01	33	F	PB	India
14	515 Aiswarya	1991-02-01	28	F	UP	India
15	502 Sridhar	1986-05-01	33	M	TN	India

### 2.COLOUR

The screenshot shows the pgAdmin 4 interface with the 'Public' schema selected. The 'colour' table is displayed in a data grid. The columns are: colour\_id and colour\_name. The data consists of 12 rows of colour names.

colour_id	colour_name
1	White
2	Black
3	Brown
4	Light Brown
5	Blue
6	Green
7	Gray
8	Amber
9	Hazel
10	Pale White
11	Moderate Brown
12	Dark Brown

### 3.FAMILY

The screenshot shows the pgAdmin 4 interface with the 'family' table selected. A query in the Query Editor window runs the command:

```
1 | SELECT * FROM family;
```

The Data Output tab displays the results:

	citizen_id	spouse_name	no_of_children
1	502	Aishwarya.R.Danush	2
2	503	Sangeeta.Vijay	2
3	504	Jessy	2
4	505	Ivana	7
5	506	Rekha.Mallya	1
6	509	Nara.Bhuvaneshwari	3
7	510	M.Natarajan	1
8	511	Sneha	2
9	512	Akshaya	1

A green success message at the bottom right indicates: "Successfully run. Total query runtime: 328 msec. 9 rows affected."

### 4.PHYSIC:-

The screenshot shows the pgAdmin 4 interface with the 'physic' table selected. A query in the Query Editor window runs the command:

```
1 | SELECT * FROM physic;
```

The Data Output tab displays the results:

	citizen_id	height	identification_mark	eye_colour	hair_colour	skin_colour
1	501	158	Mole on cheeks	2	6	3
2	502	173	White Mark on neck	2	1	1
3	503	167	Black mark in left fore hand	1	5	11
4	504	176	Mole on left hand little finger	4	9	12
5	505	179	White Mark on back	7	8	10
6	506	156	Cut on eyebrows	2	3	1
7	507	158	Mole on the right hand thumb	5	6	4
8	508	175	Cut on the nose	8	9	7
9	509	163	White Mark on thumb	11	12	10
10	510	155	Birthmark on the neck	6	5	3
11	511	166	Mole on the forehead	7	11	5
12	512	178	Mole on the eye	8	4	8
13	513	169	White Mark on the thigh	7	6	3
14	514	172	Birthmark on the right arm	4	10	11
15	515	160	Mole near the nose	2	12	9

## 5.PLACE

The screenshot shows the pgAdmin 4 interface connected to a PostgreSQL 12 database. The left sidebar shows various database objects like FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, and Tables (7). The 'Tables' section is expanded, showing citizen, colour, family, physic, place, relation, sex, Trigger Functions, Types, and Views. The 'kaps' schema is also listed. The main pane displays the 'Places' table with the following data:

	statename	states_id	country
1	Andhra Pradesh	AP	India
2	Arunachal Pradesh	AR	India
3	Assam	AS	India
4	Bihar	BR	India
5	Chhattisgarh	CG	India
6	Gujarat	GJ	India
7	Haryana	HR	India
8	Himachal Pradesh	HP	India
9	Madhya Pradesh	MP	India
10	Maharashtra	MH	India
11	Orissa	OR	India
12	Punjab	PB	India
13	Rajasthan	RJ	India
14	Tamil Nadu	TN	India
15	Uttarakhand	UK	India
16	Uttar Pradesh	UP	India
17	Chandigarh	CH	India

## 6.RELATION:-

The screenshot shows the pgAdmin 4 interface connected to a PostgreSQL 12 database. The left sidebar shows various database objects like FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, and Tables (7). The 'Tables' section is expanded, showing citizen, colour, family, physic, place, relation, sex, Trigger Functions, Types, and Views. The 'kaps' schema is also listed. The main pane displays the 'relation' table with the following data:

	citizen_id	father_name	mother_name	marital_status
1	501	R.Ramanan	R.Mythili	Unmarried
2	502	D.Dananjay	D.Ramya	Married
3	503	S.Chandrasekar	C.Shobana	Married
4	504	K.Sethuraman	S.Anjali	Married
5	505	Barron	Melania	Married
6	506	Ajay	Anjali	Married
7	507	Hirani	Indira	Unmarried
8	508	Koteswar Rao	K.Janaki Rao	Unmarried
9	509	Nara Ancham Naidu	Sameera Naidu	Married
10	510	L.Raman	R.Gayathri	Married
11	511	Aravind	Nirmala	Married
12	512	D.Doss	Raji Doss	Married
13	513	Christopher	Jennifer	Unmarried
14	514	Chubbul	Ananya	Unmarried
15	515	Ram	Ranjini	Unmarried

## 7.SEX

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects, including FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, Tables (7), Triggers, Types, and Views. A node labeled 'kaps' is expanded, showing Casts, Catalogs, Event Triggers, Extensions, and Foreign Data Wrappers. The main window contains a 'Query Editor' tab with the SQL command: 'SELECT \* FROM SEX;'. Below it is a 'Data Output' tab showing a table with two rows:

value
[PK] character varying (10)
1 M
2 F

A green message bar at the bottom right indicates: 'Successfully run. Total query runtime: 115 msec. 2 rows affected.'

## DATABASE 2:- Official DATA BASE

### 1.CR\_COLOUR:-

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects, including Event Triggers, Extensions, Foreign Data Wrappers, Languages (1) with plpgsql, and Schemas (1) with public. The public schema is expanded, showing Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, Tables (9) with commits, cr\_colour, cr\_family, cr\_physic, cr\_place, cr\_relation, cr\_sex, and crime. The main window contains a 'Query Editor' tab with the SQL command: 'SELECT \* FROM CR\_COLOUR;'. Below it is a 'Data Output' tab showing a table with 12 rows:

colour_id	colour_name
1	White
2	Black
3	Brown
4	Light Brown
5	Blue
6	Green
7	Gray
8	Amber
9	Hazel
10	Pale White
11	Moderate Brown
12	Dark Brown

## 2.CR\_FAMILY

The screenshot shows the pgAdmin 4 interface with a database management tab open. The left sidebar displays the schema structure, including Event Triggers, Extensions, Foreign Data Wrappers, Languages (1), Schemas (1), and Tables (9). The Tables section lists several tables such as commits, cr\_colour, cr\_family, cr\_physic, cr\_place, cr\_relation, cr\_sex, and cr\_crime. The main query editor window contains the following SQL query:

```
1 SELECT * FROM CR_FAMILY;
```

The Data Output tab shows the results of the query:

	criminal_id	spouse_name	no_of_children
1	605	Ivana	2
2	604	Rekha Mallya	1
3	606	M.Natarajan	1

A green message bar at the bottom right indicates: "Successfully run. Total query runtime: 164 msec. 3 rows affected".

## 3.CR\_PHYSIC

The screenshot shows the pgAdmin 4 interface with a database management tab open. The left sidebar displays the schema structure, including Event Triggers, Extensions, Foreign Data Wrappers, Languages (1), Schemas (1), and Tables (9). The Tables section lists several tables such as commits, cr\_colour, cr\_family, cr\_physic, cr\_place, cr\_relation, cr\_sex, and cr\_crime. The main query editor window contains the following SQL query:

```
1 SELECT * FROM CR_PHYSIC;
```

The Data Output tab shows the results of the query:

	criminal_id	height	identification_mark	skin_colour	eye_colour	hair_colour
1	601	158	Mole on cheeks	2	3	4
2	602	179	White Mark on back	5	6	7
3	603	156	Cut on eyebrows	8	8	10
4	604	158	Mole on the right hand thumb	11	12	6
5	605	155	Birthmark on the neck	6	8	9
6	606	172	Birthmark on the right arm	3	12	7

A green message bar at the bottom right indicates: "Successfully run. Total query runtime: 168 msec. 6 rows affected".

## 4. CR\_PLACE

The screenshot shows the pgAdmin 4 interface with the 'CRIMINAL/postgres@PostgreSQL 12' connection selected. The left sidebar displays the database schema, including 'Languages (1)' (plpgsql) and 'Schemas (1)' (public). Under the public schema, there are various objects like Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, and Tables (9). The 'Tables (9)' section is expanded, showing tables such as commits, cr\_colour, cr\_family, cr\_physic, cr\_place, cr\_relation, cr\_sex, crime, and criminal. The main pane shows the 'Data Output' tab for the 'cr\_place' table, which contains 17 rows of data:

	country	state_name	states_id
1	India	Andhra Pradesh	AP
2	India	Arunachal Pradesh	AR
3	India	Assam	AS
4	India	Bihar	BR
5	India	Chhattisgarh	CG
6	India	Gujarat	GJ
7	India	Haryana	HR
8	India	Himachal Pradesh	HP
9	India	Madhya Pradesh	MP
10	India	Maharashtra	MH
11	India	Orissa	OR
12	India	Punjab	PB
13	India	Rajasthan	RJ
14	India	Tamil Nadu	TN
15	India	Uttarakhand	UK
16	India	Uttar Pradesh	UP
17	India	Chandigarh	CH

## 5.CR\_RELATION

The screenshot shows the pgAdmin 4 interface with the 'CRIMINAL/postgres@PostgreSQL 12' connection selected. The left sidebar displays the database schema, including 'Languages (1)' (plpgsql) and 'Schemas (1)' (public). Under the public schema, there are various objects like Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, and Tables (9). The 'Tables (9)' section is expanded, showing tables such as commits, cr\_colour, cr\_family, cr\_physic, cr\_place, cr\_relation, cr\_sex, crime, and criminal. The main pane shows the 'Data Output' tab for the 'cr\_relation' table, which contains 6 rows of data:

	criminalId	father_name	mother_name	marital_status
1	601	R.Ramanan	R.Mythili	Unmarried
2	602	Barron	Melania	Married
3	603	Ajay Mallya	Anjali Mallya	Married
4	604	Hirani	Indira	Unmarried
5	605	L.Raman	R.Gayathri	Married
6	606	Chubbul Pandey	Ananya Pandey	Unmarried

## 6.CR\_SEX

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema, including the 'public' schema and a 'Tables (9)' section containing 'cr\_commits', 'cr\_colour', 'cr\_family', 'cr\_physic', 'cr\_place', 'cr\_relation', 'cr\_sex', 'crime', and 'criminal'. The main window shows the 'Query Editor' tab with the following SQL query:

```
1 SELECT * FROM CR_SEX;
```

The 'Data Output' tab displays the results of the query:

value
1 M
2 F

## 7. CRIMINAL

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema, including the 'public' schema and a 'Tables (9)' section containing 'cr\_commits', 'cr\_colour', 'cr\_family', 'cr\_physic', 'cr\_place', 'cr\_relation', 'cr\_sex', 'crime', and 'criminal'. The main window shows the 'Query Editor' tab with the following SQL query:

```
1 SELECT * FROM CRIMINAL;
```

The 'Data Output' tab displays the results of the query:

criminal_id	name	dob	age	sex	country	states_id
601	Dhanu	2000-01...	20	M	India	TN
602	Joseph	1952-08...	67	M	India	MH
603	Vijay Mallya	1959-07...	60	M	India	UP
604	Singh	1966-04...	53	M	India	UP
605	Sasi	1969-09...	50	F	India	TN
606	Phumi Pandey	1986-05...	33	F	India	PB

## 8.CRIME

The screenshot shows the pgAdmin 4 interface connected to a PostgreSQL 12 database named 'Criminal/postgres@PostgreSQL 12'. The left sidebar displays the schema browser for the 'public' schema, which includes various tables like 'commits', 'cr\_colour', 'cr\_family', etc. The main query editor window contains the following SQL query:

```
1 SELECT * FROM CRIME;
```

The resulting data output is a table with the following columns and rows:

crime_id	crime_name	details	prison_dur	
1	8001	Murder	Committing murder of a fellow...	12
2	8002	Abuse of Power	Abusing the power bestowed ...	2
3	8003	Theft	Stealing someone's belongings	1
4	8004	Attempt to murder	Attempting to murder a fellow...	1
5	8005	Destroying Public property	Destroying the property of the...	0
6	8006	Forgery	Forging and cheating people	1

## 9.COMMITS

The screenshot shows the pgAdmin 4 interface connected to a database named 'TEAM\_REPORT of dhanush.j'. The left sidebar displays the schema browser for the 'public' schema, which includes tables like 'agent', 'awards', etc. The main query editor window contains the following SQL query:

```
1 select * from commits
```

The resulting data output is a table with the following columns and rows:

criminal_id	crime_id	jail	place	date	
1	601	8005	Puzhal	TN	2019-01...
2	602	8002	Tihar	UP	2018-01...
3	605	8001	Berhampur Circle Jail	AP	2016-08...
4	606	8004	Berhampur Central Jail	AR	2012-03...
5	603	8003	Alipore Jail	PB	2016-09...
6	603	8006	Alipore Jail	PB	2017-08...
7	604	8003	Jalalabad Central Jail	UK	2016-09...
8	604	8006	Jalalabad Central Jail	UK	2017-08...

## 10. SUSPECTS

The screenshot shows the pgAdmin 4 interface with the database 'TEAM\_REPORT of dhanush' selected. The left sidebar displays the schema structure for the 'Criminal' schema, including tables like 'agent', 'awards', and 'skill'. The 'Query Editor' tab is active, showing the query: 'select \* from suspects'. The results are displayed in a Data Output grid:

	criminal_id	case_id	a_id	status
1	602	901	101	sure
2	601	907	107	doubt

## 11. AGENT DETAILS

The screenshot shows the pgAdmin 4 interface with the database 'RELATIONAL SCHEMA-conv' selected. The left sidebar displays the schema structure for the 'Official' schema, including tables like 'agent', 'awards', and 'skill'. The 'Query Editor' tab is active, showing the query: 'SELECT \* FROM AGENT;'. The results are displayed in a Data Output grid:

	agent_id	name	dob	post	year_of_service	age
1	101	James	1990-12-	Deputy Central Intelligence O...	9	29
2	102	Jake	1966-04-	Deputy Central Intelligence O...	20	53
3	103	Oliver	1975-02-	Assistant Central Intelligence ...	15	44
4	104	Harry	1991-02-	Assistant Central Intelligence ...	8	28
5	105	Charlie	2000-01-	Junior Intelligence Officer	3	20
6	106	George	1986-05-	Junior Intelligence Officer	11	33
7	107	James	1983-10-	Deputy Central Intelligence O...	16	36
8	108	William	1989-08-	Assistant Central Intelligence ...	10	30
9	109	Oscar	1952-08-	Junior Intelligence Officer	20	67
10	110	Thomas	1969-09-	Deputy Central Intelligence O...	18	50

Message bar: ✓ Successfully run. Total query runtime: 203 msec. 10 rows affected.

## 12.POST OF AGENT

The screenshot shows the pgAdmin 4 interface with the following details:

- Toolbar:** Includes File, Object, Tools, Help, and various database management icons.
- Header:** Shows the connection details "Official/postgres@PostgreSQL 12" and the port "127.0.0.1:52256/browser".
- Left Panel (Browser):** Displays the schema tree under "Official", including FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, Tables (12), Trigger Functions, Types, and Views. The "Tables (12)" section is expanded, showing ag\_post, ag\_skill, agent, awards, case\_assign, case\_det, hd\_post, head, login\_agent, login\_head, skill, and year\_of\_service.
- Central Panel (Query Editor):** Contains the SQL query: `SELECT * FROM AG_POST;`. The results are displayed in a Data Output grid:

name
[PK] character varying (40)
1 Deputy Central Intelligence Officer
2 Junior Intelligence Officer
3 Assistant Central Intelligence Offi...

- Status Bar:** Shows a green success message: "Successfully run. Total query runtime: 202 msec. 3 rows affected." and the system status bar at the bottom.

## 13.AGENT SKILL

The screenshot shows the pgAdmin 4 interface with the following details:

- Toolbar:** Includes File, Object, Tools, Help, and various database management icons.
- Header:** Shows the connection details "Official/postgres@PostgreSQL 12" and the port "127.0.0.1:52256/browser".
- Left Panel (Browser):** Displays the schema tree under "Official", including FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, Tables (12), Trigger Functions, Types, and Views. The "Tables (12)" section is expanded, showing ag\_post, ag\_skill, agent, awards, case\_assign, case\_det, hd\_post, head, login\_agent, login\_head, skill, and year\_of\_service.
- Central Panel (Query Editor):** Contains the SQL query: `SELECT * FROM AG_SKILL;`. The results are displayed in a Data Output grid:

agent_id	skill
[PK] numeric (10)	[PK] character varying (100)
1	101 Investigating
2	101 catching thieves
3	102 Investigating
4	102 spying
5	103 Investigating
6	104 Investigating
7	104 catching thieves
8	105 Investigating
9	105 murder solving
10	106 Investigating
11	106 catching thieves
12	107 Investigating
13	107 theft solving
14	108 Investigating
15	108 murder solving
16	109 Investigating
17	109 catching thieves
18	110 Investigating

- Status Bar:** Shows the system status bar at the bottom.

## 14.AWARDS

The screenshot shows the pgAdmin 4 interface with the title bar "RELATIONAL SCHEMA-conv" and "DATABASE MANAGEMENT" and the window title "pgAdmin 4". The browser pane on the left lists various database objects under "Tables (12)". The main pane displays the results of the query "SELECT \* FROM AWARDS;". The table has columns: agent\_id, case\_id, and award. The data output is as follows:

	agent_id	case_id	award
1		101	901 presidential rank award
2		102	902 national award fro distinguished ex...
3		103	903 national award fro public service
4		104	904 intelligence star award
5		105	905 national award for excellent investi...
6		107	907 20 years of excellence
7		106	906 presidential rank award
8		108	908 national award fro distinguished ex...
9		109	909 national award for excellent investi...
10		110	910 intelligence star award

A message at the bottom right indicates: "Successfully run. Total query runtime: 190 msec. 10 rows affected".

## 15.CASE\_ASSIGN

The screenshot shows the pgAdmin 4 interface with the title bar "TEAM\_REPORT of dhanush" and "pgAdmin 4". The browser pane on the left lists various database objects under "Tables (7)" and "Triggers". The main pane displays the results of the query "select \* from case\_assign". The table has columns: case\_id, head\_id, agent\_id, date\_of\_comp, status, deadline, and date\_of\_assign. The data output is as follows:

	case_id	head_id	agent_id	date_of_comp	status	deadline	date_of_assign
1	907	1002	107	2018-09-17	completed	2018-10-17	2018-07-17
2	908	1002	108	2018-07-16	completed	2018-09-16	2018-06-16
3	909	1002	109	2019-01-20	completed	2019-02-21	2019-01-10
4	905	1001	105	[null]	undergoing	2020-03-01	2020-01-10
5	906	1002	106	[null]	new	2020-04-01	2020-02-02
6	901	1001	101	[null]	undergoing	2020-04-05	2020-03-03
7	904	1001	104	[null]	undergoing	2020-02-20	2019-01-20
8	902	1001	102	[null]	undergoing	2020-03-01	2020-01-20
9	903	1001	103	[null]	undergoing	2020-04-10	2020-02-29
10	910	1002	109	2020-03-30	completed	2019-12-12	2019-09-09

## 16.CASE\_DETAILS

The screenshot shows the pgAdmin 4 interface with the database 'TEAM\_REPORT' selected. In the left sidebar, under the 'case\_det' table, the 'Triggers' section is expanded. A query in the 'Query Editor' window is running:

```
1 select * from case_det
```

The results are displayed in a table titled 'Data Output':

case_id	case_name	details	specialisation	date_of_hp	place
1	901 Alcatraz escape	Capturing three men who esc...	catching thieves	2017-01-29	UP
2	902 Spy	spying the Russians	spying	2020-02-18	AP
3	903 Bombing	investigating bombing of the ...	investigating	2019-03-25	BR
4	904 D B cooper hijacking	Investigating about the man ...	catching thieves	2020-04-05	CG
5	905 Murder of workers	Investigating the murder of th...	solving murder	2018-05-10	GJ
6	906 Hijack in Bombay	Investigating the hijack of a pl...	solving hijack	2017-06-11	MH
7	907 Robbery in bank	Catching the thief who robbe...	theft solving	2016-07-28	OR
8	908 Murder of ex-minister	Unravelling the murder of an e...	solving murder	2019-08-18	PB
9	909 Escape of thief	Finding the thief who escaped...	catching thief	2019-09-30	CH
10	910 9/11 investigation	Investigation and other activit...	investigating	2018-10-09	UK

A green message bar at the bottom right indicates: "Successfully run. Total query runtime: 911 msec. 10 rows affected".

## 17.HEAD DETAILS

The screenshot shows the pgAdmin 4 interface with the database 'RELATIONAL SCHEMA-con' selected. In the left sidebar, under the 'HEAD' table, the 'Tables' section is expanded. A query in the 'Query Editor' window is running:

```
1 SELECT * FROM HEAD;
```

The results are displayed in a table titled 'Data Output':

head_id	name	dob	post	year_of_service	age
1	Kaplesh	1966-04-15	Deputy Director	20	53
2	Aswin	1969-09-10	Assistant Deputy Director	19	50

A green message bar at the bottom right indicates: "Successfully run. Total query runtime: 183 msec. 2 rows affected".

## 18.HEAD POST

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects, including FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, and Tables (12). The Tables section is expanded, showing ag\_post, ag\_skill, agent, awards, case\_assign, case\_det, hd\_post, head, login\_agent, login\_head, skill, and year\_of\_service. The main query editor window contains the following SQL code:

```
1 SELECT * FROM HD_POST;
```

The Data Output tab shows the results of the query:

name
[PK] character varying (40)
1 Deputy Director
2 Assistant Deputy Director

A status bar at the bottom right indicates: Successfully run. Total query runtime: 168 msec. 2 rows affected.

## 19.LOGIN OF AGENT

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects, including FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, and Tables (12). The Tables section is expanded, showing ag\_post, ag\_skill, agent, awards, case\_assign, case\_det, hd\_post, head, login\_agent, login\_head, skill, and year\_of\_service. The main query editor window contains the following SQL code:

```
1 SELECT * FROM LOGIN_AGENT;
```

The Data Output tab shows the results of the query:

agent_id	login_id	password
[PK] numeric (10)	character varying (20)	character varying (20)
1	jamesia.in	jame
2	jakesia.in	jake
3	oliversia.in	oliver
4	harrysia.in	harry
5	charliesia.in	charlie
6	georgesia.in	george
7	jamessia.in	james
8	williamsia.in	william
9	oscarzia.in	oscar
10	thmassia.in	thomas

A status bar at the bottom right indicates: Successfully run. Total query runtime: 154 msec. 10 rows affected.

## 20.LOGIN OF HEAD

The screenshot shows the pgAdmin 4 interface with the following details:

- Toolbar:** Includes File, Object, Tools, Help, and a search bar.
- Left Sidebar (Browser):** Shows the database schema with a tree view of tables, functions, and other objects.
- Central Area:** A "Query Editor" window displays the SQL query: `SELECT * FROM LOGIN_HEAD;`. Below it is a "Data Output" window showing the results of the query:

head_id	login_id	password
1	1001	kapsguru3033
2	1002	aswin

- Status Bar:** Shows a green success message: "Successfully run. Total query runtime: 140 msec. 2 rows affected".
- Taskbar:** Shows the Windows taskbar with various pinned icons like File Explorer, Edge, and File.

## 21.SKILL

The screenshot shows the pgAdmin 4 interface with the following details:

- Toolbar:** Includes File, Object, Tools, Help, and a search bar.
- Left Sidebar (Browser):** Shows the database schema with a tree view of tables, functions, and other objects.
- Central Area:** A "Query Editor" window displays the SQL query: `SELECT * FROM SKILL;`. Below it is a "Data Output" window showing the results of the query:

name
catching thieves
Investigating
murder solving
spying
theft solving

- Status Bar:** Shows a green success message: "Successfully run. Total query runtime: 162 msec. 5 rows affected".
- Taskbar:** Shows the Windows taskbar with various pinned icons like File Explorer, Edge, and File.

## 22.YEAR\_OF\_SERVICE

The screenshot shows the pgAdmin 4 interface with the 'Data Output' tab selected. The query results are displayed in a table:

value	[PK] numeric (4)
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
--	--

## 23. AGENT\_SEEK:-

The screenshot shows the pgAdmin 4 interface with the 'Data Output' tab selected. The query results are displayed in a table:

s1_ag_id	s2_ag_id	status	case_id
[PK] numeric (10)	[PK] numeric (10)	character varying (10)	integer
1	101	107 allowed	907

# 1. SQL QUERIES

S.NO.	NAME	TABLES INVOLVED	CONSTRUCTS USED
1.	PRISON DURATION OF CRIMINALS	CRIMINAL,COMMITS,CRIME	NATURAL JOIN, GROUPING

## DESCRIPTION:-

The query is used to list the no.of years each prisoners in the database has spent.

## Query:-

```
select criminal_id,name,sum(prison_dur) from criminal natural join commits natural join crime group by criminal_id
```

```
1 select criminal_id, name, sum(prison_dur) from criminal natural join commits natural join crime group by criminal_id
```

criminal_id	name	sum
605	Sasi	12
602	Joseph	2
604	Singh	2
601	Dhanu	0
603	Vijay Mallu	2
606	Phumi Pandey	1

Successfully run. Total query runtime: 844 msec. 6 rows affected

2.	STATE WITH HIGHEST NO. OF CRIMINALS	COMMITS	NATURAL JOIN
----	-------------------------------------	---------	--------------

## QUERY:-

```
select * from cr_place natural join (select place as states_id,count from ((select place,count(distinct crime_id) from commits group by place) as temp3 natural join (select max(count) as count from (select place,count(distinct crime_id) as count from commits group by place) as temp) as temp2) as temp4) as temp1
```

The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1  ant from (select place,count(distinct crime_id) as count from commits group by place) as temp) as temp2) as temp4) as temp1
```

The Data Output pane shows the results:

states_id	country	state_name	count
PB	India	Punjab	2
UK	India	Uttarakhand	2

Message: Successfully run. Total query runtime: 968 msec. 2 rows affected.

3.	<b>NO. OF CASES HANDLED BY EACH AGENT ALONG WITH STATUS</b>	<b>CASE_ASSIGN</b>	<b>AGGREGATE FUNCTION, GROUPING</b>
----	---	--------------------	---

## QUERY:-

```
select agent_id,status,count(case_id) from case_assign group by agent_id,status
```

The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1  select agent_id,status,count(case_id) from case_assign group by agent_id,status
```

The Data Output pane shows the results:

agent_id	status	count
108	completed	1
109	completed	2
104	undergoing	1
106	new	1
101	undergoing	1
103	undergoing	1
102	undergoing	1
107	completed	1
105	undergoing	1

4.	NO. OF AGENTS WORKING IN EACH POST	AG_POST,AGENT	LEFT OUTER JOIN, GROUPING
----	------------------------------------	---------------	------------------------------

QUERY:-

```
select post,count(agent_id) from ag_post left outer join agent on ag_post.name=agent.post group by post
```

```
1 select post, count(agent_id) from ag_post left outer join agent on ag_post.name=agent.post group by post
```

post	count
Assistant Central Intelligence...	3
Junior Intelligence Officer	4
Deputy Central Intelligence ...	4

Successfully run. Total query runtime: 1 secs 239 msec. 3 rows affected

5.	NO. OF AWARDS WON BY VARIOUS AGENTS IN THE CURRENT YEAR	AWARDS	NATURAL JOIN, DATE FUNCTIONS
----	---	--------	---------------------------------

QUERY:-

```
select * from (select agent_id,case_id,award,extract (year from dt_of_award) as year from awards) as temp
natural join (select extract(year from current_date) as year) as temp1
```

```
1 select * from (select agent_id, case_id, award, extract (year from dt_of_award) as year from
2 awards) as temp
3 natural join (select extract(year from current_date) as year) as temp1
```

year	agent_id	case_id	award
2020	101	901	presidential rank award
2020	102	902	national award for distinguished ex...
2020	103	903	national award for public service
2020	104	904	intelligence star award
2020	105	905	national award for excellent investi...
2020	107	907	20 years of excellence

6.	NO. OF CASES ASSIGNED BY A HEAD OFFICIAL EACH MONTH IN A PARTICULAR YEAR	CASE_ASSIGN	DATE FUNCTIONS
----	--	-------------	----------------

QUERY:-

```
select extract(month from date_of_assign) as mon, count(extract(month from date_of_assign)) from
case_assign where head_id=1001 and extract(year from date_of_assign)=2020 group by mon
```

The screenshot shows the pgAdmin 4 interface. On the left is the 'Browser' pane displaying the database schema with tables like 'skill', 'agent', and 'awards'. The 'Query Editor' pane contains the following SQL query:

```
select extract(month from date_of_assign) as mon, count(extract(month from date_of_assign)) from case_assign where head_id=1001 and extract(year from date_of_assign)=2020 group by mon
```

The 'Data Output' pane shows the results of the query:

mon	count
1	2
2	1
3	2

A message at the bottom right indicates: "Successfully run. Total query runtime: 1 secs 917 msec. 3 rows affected".

7.	CRIMINAL WHO HAS SPENT THE HIGHEST NO OF YEARS IN PRISON	CRIMINAL,COMMITS	NATURAL JOIN, AGGREGATE FUNCTIONS , SUBQUERIES
----	---	------------------	---

QUERY:-

```
select * from (select criminal_id, name, sum(prison_dur) as count from criminal natural join commits natural
join crime group by criminal_id) as temp2 natural join (select max(sum) as count from (select
criminal_id, name, sum(prison_dur) from criminal natural join commits natural join crime group by
criminal_id) as temp) as temp3
```

```

1 select * from (select criminal_id, name, sum(prison_dur) as count from criminal natural join
2 commits natural join crime group by criminal_id) as temp2 natural join
3 (select max(count) as count from (select criminal_id, name, sum(prison_dur) from criminal natural join
4 commits natural join crime group by criminal_id) as temp) as temp3

```

Successfully run. Total query runtime: 1 secs 246 msec. 1 rows affected

8.	QUERING THE PUBLIC TABLE FOR A NAME	CITIZEN,RELATION, PHYSIC,PLACE	LIKE OPERATOR,NATURAL JOIN
----	---	-----------------------------------	----------------------------------

QUERY:-

```
select citizen_id, name, age, sex, statename, country, father_name, height, identification_mark from citizen
natural join relation natural join physic natural join place where name like '%Se%'
```

citizen_id	name	age	sex	statename	country	father_name	height	identification_mark
504	Sethu	36	M	Tamil Nadu	India	K.Sethuraman	176	Mole on left hand lif

9.	NO. OF CASES COMPLETED BETWEEN A GIVEN INTERVAL OF TIME THAT WERE ASSIGNED BY A HEAD OFFICIAL	CASE_ASSIGN,CASE_DET	NATURAL JOIN
----	---	----------------------	--------------

QUERY:-

```
select case_id,case_name from case_assign natural join case_det where date_of_assign>='2019-02-02'
```

```
and date_of_assign<='2020-02-02' and head_id=1002 and status='completed'
```

The screenshot shows the pgAdmin 4 interface with a query result displayed in the 'Query Editor' tab. The query executed was:

```
select case_id,case_name from case_assign natural join case_det where date_of_assign>='2019-02-02'  
and date_of_assign<='2020-02-02' and head_id=1002 and status='completed'
```

The result set contains one row:

case_id	case_name
910	9/11 investigation

10.	AGENT WHO HAS SOLVED THE MAXIMUM NO . OF CASES	CASE_ASSIGN	NATURAL JOIN,AGGREGATE FUNCTION
-----	--	-------------	---------------------------------

QUERY:-

```
select agent_id,name,count as max from (select agent_id,count(case_id) from case_assign group by agent_id) as temp1 natural join(select max(count) as count from (select agent_id,count(case_id) from case_assign group by agent_id) as temp) as temp2 natural join agent
```

agent_id	name	max
109	Oscar	2

# VIEWS

S.NO.	NAME	TABLES	DESCRIPTION
1.	CRIMINAL_DET	CRIMINAL,COMMITS,CRIME	LISTING THE CRIMES COMMITTED BY EACH CRIMINAL

QUERY:-

```
create view criminal_det as select criminal_id,name,dob,age,sex,state_name,country,crime_name,details
from criminal natural join commits natural join crime natural join cr_place
```

The screenshot shows the pgAdmin 4 interface. In the top navigation bar, 'File', 'Object', 'Tools', and 'Help' are visible. Below the toolbar, there's a search bar and a dropdown menu for '127.0.0.1:53828/browser/'. The main window has tabs for 'Dashboard', 'Properties', 'SQL', 'Statistics', 'Dependencies', 'Dependents', 'Criminal/postgre...', 'public.case\_ass...', 'Criminal/postre...', 'public.case\_ass...', and 'Criminal/post...'. A 'Query History' tab is selected, showing a single entry: 'Recorded t 1 SELECT \* FROM CRIMINAL\_DET'. Below this, a 'Data Output' tab is active, displaying the results of the query. The results are presented in a table with columns: criminal\_id, name, dob, age, sex, state\_name, country, crime\_name, and details. The data shows 10 entries for various criminals with their details like name, date of birth, age, sex, state, country, committed crime, and details about the crime.

criminal_id	name	dob	age	sex	state_name	country	crime_name	details
601	Dhanu	2000-01-01	20	M	Tamil Nadu	India	Destroying Public property	Destroying the property
602	Joseph	1952-08-01	67	M	Maharashtra	India	Abuse of Power	Abusing the power be
605	Sasi	1969-09-01	50	F	Tamil Nadu	India	Murder	Committing murder o
606	Phumi Pandey	1986-05-01	33	F	Punjab	India	Attempt to murder	Attempting to murder
603	Vijay Mallya	1959-07-01	60	M	Uttar Pradesh	India	Theft	Stealing someone's be
603	Vijay Mallya	1959-07-01	60	M	Uttar Pradesh	India	Forgery	Forging and cheating
604	Singh	1966-04-01	53	M	Uttar Pradesh	India	Theft	Stealing someone's be
604	Singh	1966-04-01	53	M	Uttar Pradesh	India	Forgery	Forging and cheating

2.	CIT_DET	CITIZEN,FAMILY,RELATION,PLACE	TO LIST THE PERSONAL DETAILS OF PUBLIC
----	---------	-------------------------------	--

## QUERY:-

```
create view cit_det as select * from citizen natural join family natural join relation natural join place
```

The screenshot displays two windows of the pgAdmin 4 interface, both showing the results of the same SQL query. The top window shows the initial view 'cit\_det' with the following schema:

states_id	country	citizen_id	name	dob	age	sex	spouse_name	no_of_children
TN	India	502	Sridhar	1986-05-15	33	M	Aishwarya.R.Danush	
TN	India	503	Vijay	1978-06-10	41	M	Sangeeta Vijay	
TN	India	504	Sethu	1983-10-20	36	M	Jessy	
MH	India	505	Donald	1952-08-15	67	M	Ivana	
UP	India	506	Mallya	1959-07-20	60	M	Rekha Mallya	
AP	India	509	Chandrababu	1951-03-15	68	M	Nara Bhuvaneswari	
TN	India	510	kala	1969-09-10	50	F	M.Natarajan	
AP	India	511	Arjun	1989-08-15	30	M	Sneha	
TN	India	512	Karthikeyan	1989-11-15	30	M	Akshaya	

The bottom window shows the expanded view with the following schema:

sex	spouse_name	no_of_children	father_name	mother_name	marital_status	statename
33 M	Aishwarya.R.Danush		2 D.Dananjayan	D.Ramya	Married	Tamil Nadu
41 M	Sangeeta Vijay		2 S.Chandrasekar	C.Shobana	Married	Tamil Nadu
36 M	Jessy		2 K.Sethuraman	S.Anjali	Married	Tamil Nadu
67 M	Ivana		7 Barron	Melania	Married	Maharashtra
60 M	Rekha Mallya		1 Ajay	Anjali	Married	Uttar Pradesh
68 M	Nara Bhuvaneswari		3 Nara Ancham Naidu	Sameera Naidu	Married	Andhra Pradesh
50 F	M.Natarajan		1 L.Raman	R.Gayathri	Married	Tamil Nadu
30 M	Sneha		2 Aravind	Nirmala	Married	Andhra Pradesh
30 M	Akshaya		1 D.Doss	Raji Doss	Married	Tamil Nadu

3.	YR_CS	CASE_ASSIGN	NO. OF CASES COMPLETED BY THE AGENCY EACH YEAR
----	-------	-------------	---

QUERY:-

```
create view yr_cs as select extract(year from date_of_assign) as yr, count(case_id) from
case_assign where status='completed' group by yr
```

The screenshot shows the pgAdmin 4 interface. On the left is the object browser tree under the schema 'wser'. It lists several objects: 'hd\_det(h\_id num')', 'Materialized Views', 'Procedures (5)', 'Sequences', 'Tables (24)', and 'ag\_seek' (which is currently selected). Under 'Tables (24)', there are 'ag\_post' and 'ag\_seek'. The 'ag\_seek' table has four columns: 'case\_id', 's1\_ag\_id', 's2\_ag\_id', and 'status'. The 'Query Editor' tab is active, displaying the SQL query: 'SELECT \* FROM YR\_CS'. Below the query is the 'Data Output' pane, which shows the results of the query. The results are as follows:

yr	count
2018	2
2019	2

A green success message at the bottom right of the data output pane states: 'Successfully run. Total query runtime: 1 secs 622 msec. 2 rows affected.'

# FUNCTIONS

S.NO.	NAME	TABLES_INVOLVED	PARAMETERS	RETURN TYPE
1.	CAS_DET	SUSPECTS,CASE_ASSIGN, AGENT,HEAD	CASE_ID	returns table( ag_id numeric, ag_name varchar, hd_id numeric, h_name varchar, cas_name varchar, cas_det varchar )

This function is used to display the various details about a particular case

QUERY:-

```

create or replace function cas_det(cas_id numeric)

returns table(
    ag_id numeric, ag_name varchar, hd_id numeric, h_name varchar, cas_name varchar, cas_det varchar
)

as $$

declare

t_temp record;

det cursor for select case_id, case_name, details, agent_id, name, head_id, hd_name, count as no_of_suspects from (select
case_id, case_name, details, agent_id, name, count, head_id, date, place from (select case_id, count(criminal_id) from
suspects group by case_id) as temp natural join case_assign natural join case_det natural join agent) as temp2 natural
join head where case_id=cas_id;

begin

open det;

```

```

loop
fetch from det into t_temp;exit when not found;

ag_id:=t_temp.agent_id;ag_name:=t_temp.name;hd_id:=t_temp.head_id;h_name:=t_temp.hd_name;cas_name:=t_temp.
case_name;cas_det:=t_temp.details;

return next;

end loop;

end $$

language plpgsql

```

ag_id	ag_name	hd_id	h_name	cas_name	cas_det
101	James	1001	Kapilesh	Alcatraz escape	Capturing three men w...

2.	HD_DET	CASE_ASSIGN, AGENT, CASE_DET	HEAD_ID	returns table( c_id numeric,c_name varchar, detai varchar, a_id numeric, a_name varchar, days numeric,status varchar )
----	--------	------------------------------------	---------	--

This function is used to list various data about an head official from the database

**Query:-**

```
create or replace function hd_det(h_id numeric)
returns table(
c_id numeric,c_name varchar,detai varchar,a_id numeric,a_name varchar,days numeric,status varchar
)
as $$

declare
t_temp record;

det cursor for select case_id,case_name,details,agent_id,name,count from (select
case_id,agent_id,(current_date-deadline) as count from case_assign where head_id=1001) as temp
natural join case_det natural join agent;

begin

open det;

loop
fetch from det into t_temp;
exit when not found;

c_id:=t_temp.case_id;c_name:=t_temp.case_name;detai:=t_temp.details;a_id:=t_temp.agent_id;
a_name:=t_temp.agent_id;

if t_temp.count<0 then
    days:=t_temp.count*(-1);
    status='past deadline';
else

```

```

days:=t_temp.count;

status:='time available';

end if;

return next;end loop;close det;

end $$

language plpgsql

```

3.	AG_ID_GEN	agent	No parameter	AN INTEGER
----	-----------	-------	--------------	------------

This function is used to generate a new agent id

QUERY:-

```

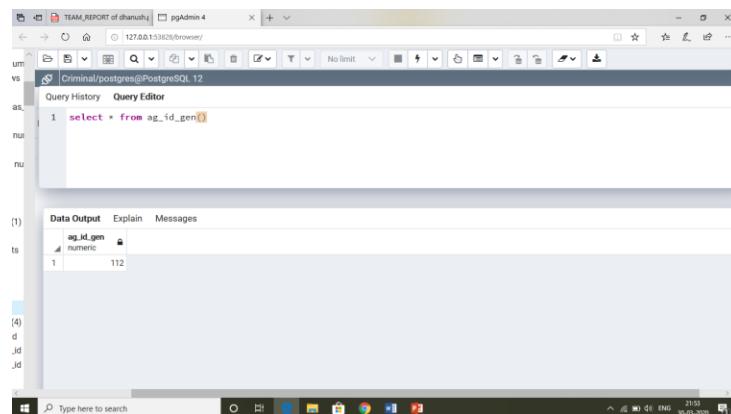
create or replace function ag_id_gen()
returns numeric
as $$

declare
v_temp numeric;

begin
select max(agent_id) into v_temp from agent;v_temp:=v_temp+1;return v_temp;
end $$

language plpgsql

```



4.	AG_DE	AGENT,CASE_ASSIGN, CASE_DET,HEAD	AGENT ID AND STATUS	returns table( n varchar, c_id numeric, c_name varchar, det varchar(100), h_id numeric, h_name varchar )
----	-------	-------------------------------------	------------------------	--

This function is used to list the agent and case details with a corresponding status.

Hence , this function can be invoked by 2 webpage : a page which needs to display all details about the undergoing agent of that agent and another page which lists all the completed cases of the agent

QUERY:-

```
create or replace function ag_de(ag_id numeric,stat varchar)
returns table(
n varchar,
c_id numeric,
c_name varchar,
det varchar(100),
h_id numeric,
h_name varchar,
day_left numeric)
```

```

)
as $$

declare
t_day numeric;
t_temp record;

de cursor for select agent_id,name,case_id,case_name,details,temp.head_id,hd_name,date_of_hp from
(select * from agent natural join case_assign natural join case_det) as temp,head where
temp.head_id=head.head_id order by date_of_hp asc;

begin
open de;
loop
fetch from de into t_temp;
exit when not found;
if t_temp.agent_id=ag_id and t_temp.status=stat then
n:=t_temp.name;
t_day:=current_date-t_temp.date_of_hp;
day_left:=t_day;
c_id:=t_temp.case_id;c_name:=t_temp.case_name;
det:=t_temp.details; h_id:=t_temp.head_id;h_name:=t_temp.hd_name;
return next;
end if;
end loop;
end;$$

language plpgsql

```

```

1 select * from AG_DE(102, 'undergoing')

(1)
ts
n character varying | c_id numeric | c_name character varying | det character varying | h_id numeric | h_name character varying |
1 Jake | 902 | Spy | spying the Russians | 1001 | Kapilesh

```

## PROCEDURES

S.NO.	NAME	TABLES_INVOLVED	PARAMETERS
1.	COMP_CASE	CASE_ASSIGN,HD_SEEK	CASE_ID(IN)

When a case is completed , the procedure comp\_case is called. It changes the status,updates the completion date and deletes any public database requests made by on behalf of the case.

QUERY:-

```

create or replace procedure comp_case(cas_id integer)
language plpgsql
as $$

begin

update case_assign set date_of_comp=current_date,status='completed' where case_id=cas_id;

delete from hd_seek where case_id=cas_id;

end;$$

```

TEAM\_REPORT of dhanushj pgAdmin 4 127.0.0.1:53828/browser/

```
Query History Query Editor
as.
nui.
nui.

(1) Data Output Explain Messages
ts
  case_id [PK] numeric (10) head_id [PK] numeric (10) agent_id [PK] numeric (10) date_of_comp date status character varying (20) deadline date date_of_assign
  1 902 1001 102 [null] undergoing 2020-03-01 2020-01-20
  2 903 1001 103 [null] undergoing 2020-04-10 2020-02-29
  3 907 1002 107 2018-09-17 completed 2018-10-17 2018-07-17
  4 908 1002 108 2018-07-16 completed 2018-09-16 2018-06-16
  5 909 1002 109 2019-01-20 completed 2019-02-21 2019-01-10
  6 905 1001 105 [null] undergoing 2020-03-01 2020-01-10
  7 906 1002 106 [null] new 2020-04-05 2020-02-02
  8 901 1001 101 [null] undergoing 2020-04-05 2020-03-03
  9 904 1001 104 [null] undergoing 2020-02-20 2019-01-20
  10 910 1002 109 2020-03-30 completed 2019-12-12 2019-09-09
```

TEAM\_REPORT of dhanushj pgAdmin 4 127.0.0.1:53828/browser/

```
Dashboard Properties SQL Statistics Dependencies Dependents Criminal/postgre... public.case_ass... Criminal/postgre... public.case_...
um.
vs.
Criminal/postgres@PostgreSQL 12
Query History Query Editor
as.
nui.
nui.

(1) Data Output Explain Messages
ts
  ag_id [PK] numeric (10) hd_id [PK] numeric (10) status character varying (10) case_id integer
  1 101 1001 new 1001
  2 104 1001 granted 904
```

TEAM\_REPORT of dhanushj pgAdmin 4 127.0.0.1:53828/browser/

```
Query History Query Editor
as.
nui.
nui.

(1) Data Output Explain Messages
ts
  case_id [PK] numeric (10) head_id [PK] numeric (10) agent_id [PK] numeric (10) date_of_comp date status character varying (20) deadline date date_of_assign
  1 902 1001 102 [null] undergoing 2020-03-01 2020-01-20
  2 903 1001 103 [null] undergoing 2020-02-29
  3 904 1001 104 2020-03-30 completed 2020-02-20 2019-01-20
  4 907 1002 107 2018-09-17 completed 2018-10-17 2018-07-17
  5 908 1002 108 2018-07-16 completed 2018-09-16 2018-06-16
  6 909 1002 109 2019-01-20 completed 2019-02-21 2019-01-10
  7 905 1001 105 [null] undergoing 2020-03-01 2020-01-10
  8 906 1002 106 [null] new 2020-04-05 2020-02-02
  9 901 1001 101 [null] undergoing 2020-04-05 2020-03-03
  10 910 1002 109 2020-03-30 completed 2019-12-12 2019-09-09
```

```

1
2 select * from hd_seek

```

	ag_id	hd_id	status	case_id
1	101	1001	new	1001

Successfully run. Total query runtime: 1 secs 793 msec. 1 rows affect

2.	Sus_det	Case_det,suspect,criminal	CRIMINAL_ID(IN)
----	---------	---------------------------	-----------------

This procedure is used to list the case details fro which a criminal has been a suspect . Seeing this , an agent can make a request to another agent who has the case details of any particular case.

QUERY:-

```

create or replace procedure sus_det (cr_id numeric)
language plpgsql
as $$

declare

det cursor for select * from suspects natural join criminal natural join case_det where criminal_id=cr_id;

t_temp record;

begin

open det;

loop

fetch from det into t_temp;

```

```

exit when not found;

raise notice 'case_id:%;case_name:%;criminal_id:%;criminal_name:%',t_temp.case_id,t_temp.case_name,
t_temp.criminal_id,t_temp.name;

end loop;

close det;

open det;

loop

fetch from det into t_temp;

exit when not found;

if t_temp.status='sure' then

raise notice 'Main criminal of % % is %
%',t_temp.case_id,t_temp.case_name,t_temp.criminal_id,t_temp.name;

end if;

end loop;

close det;

end $$
```

The screenshot shows a pgAdmin 4 interface with a query editor window. The query is:

```
call sus_det(602)
```

The 'Messages' tab displays the following output:

```

NOTICE: case_id:901;case_name:Alcatraz escape;criminal_id:602;criminal_name:Joseph
NOTICE: Main criminal of 901 Alcatraz escape is 602 Joseph
CALL

Query returned successfully in 1 secs 669 msec.
```

3.	AG_SK	AG_SEEK,CASE_DET,SUSPECTS	AGENT1_ID(IN), AGENT2_ID(IN), COURSE_ID(IN)
----	-------	---------------------------	---

This procedure enables an agent1 to seek another agent2 to get the details of a course\_id.

QUERY:-

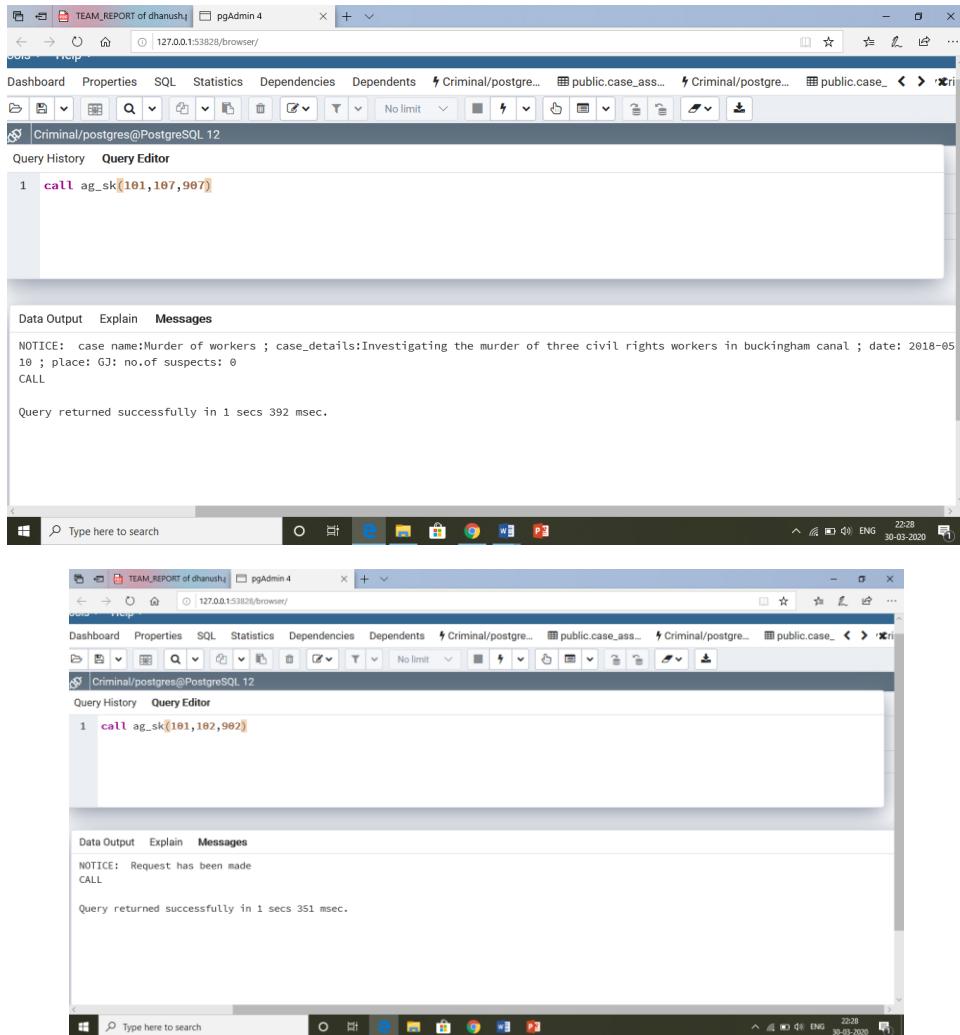
```
create or replace procedure ag_sk(agent1_id numeric,agent2_id numeric,c_id numeric)
language plpgsql
as $$

declare
t_temp record;
v_st varchar;

begin
select status into v_st from ag_seek where s1_ag_id=agent1_id and s2_ag_id=agent2_id and case_id=c_id;
if v_st='new' then
    raise notice 'still not granted';
elsif v_st='allowed' then
    select case_name,details,date_of_hp,place,count(criminal_id) into t_temp from case_det left outer
join suspects on case_det.
case_id=suspects.case_id group by case_det.case_id;      raise notice 'case name:% ; case_details:% ; date:
% ; place: %: no.of suspects: %',t_temp.case_name,t_temp.details
,t_temp.date_of_hp,t_temp.place,t_temp.count;
Else insert into ag_seek(s1_ag_id,s2_ag_id,case_id,status) values(agent1_id,agent2_id,c_id,'new');
```

```
raise notice 'Request has been made';

end if;end $$
```



The image shows two separate instances of the pgAdmin 4 interface, each displaying a query editor window. Both windows have the title 'TEAM\_REPORT of dhanush' and are connected to the database 'Criminal/postgres@PostgreSQL 12'. The query in both editors is:

```
1 call ag_sk(101,107,907)
```

The results pane below the query editor shows the following output for both sessions:

NOTICE: case name:Murder of workers ; case\_details:Investigating the murder of three civil rights workers in buckingham canal ; date: 2018-05-10 ; place: GJ; no.of suspects: 0  
CALL

Query returned successfully in 1 secs 392 msec.

Below the results pane, the status bar indicates the time as 22:28 and the date as 30-03-2020.

# TRIGGERS

S.NO.	NAME	DESCRIPTION	TYPE	TABLES INVOLVED
1.	Comp_star	Used to assign stars to an agent once he completes his mission. Additional 5 stars are given if he/she finishes before deadline	After Update	Case_assign,Agent

QUERY:-

```
create or replace trigger comp_star
after update on case_assign
for each row
execute procedure prod_star();
```

```
create or replace function prod_star()
```

```
returns trigger as
```

```
$$
```

```
declare
```

```
v_star numeric;
```

```
begin
```

```
if new.status='completed' and old.status='undergoing' then
```

```

select star into v_star from agent where agent_id=old.agent_id;raise notice 'value:%',old.agent_id;

v_star:=v_star+5;

if new.date_of_comp<old.deadline then

    v_star:=v_star+5;

end if;

update agent set star=v_star where agent_id=old.agent_id;

end if;

return new;end $$

language plpgsql

```

Screenshot of pgAdmin 4 showing the results of a query on the 'case\_assign' table.

**Data Output**

case_id	head_id	agent_id	date_of_comp	status	deadline	date_of_assign
1	902	1001	102 [null]	undergoing	2020-03-01	2020-01-20
2	903	1001	103 [null]	undergoing	2020-04-10	2020-02-29
3	904	1001	104 2020-03-30	completed	2020-02-20	2019-01-20
4	907	1002	107 2018-07-17	completed	2018-10-17	2018-07-17
5	908	1002	108 2018-07-16	completed	2018-09-16	2018-06-16
6	909	1002	109 2019-01-20	completed	2019-02-21	2019-01-10
7	905	1001	105 [null]	undergoing	2020-03-01	2020-01-10
8	906	1002	106 [null]	new	2020-04-01	2020-02-02
9	901	1001	101 [null]	undergoing	2020-04-05	2020-03-03
10	910	1002	109 2020-03-30	completed	2019-12-12	2019-09-09

Screenshot of pgAdmin 4 showing the results of a query on the 'agent' table.

**Data Output**

agent_id	name	dob	post	year_of_service	age	star
1	107 Charles	1983-10-10	Deputy Central Intelligence...	16	36	0
2	102 Jake	1966-04-15	Deputy Central Intelligence...	20	53	5
3	110 Thomas	1969-09-20	Deputy Central Intelligence...	18	50	5
4	103 Oliver	1975-02-15	Assistant Central Intelligenc...	15	44	10
5	111 kapil	2002-02-15	Junior Intelligence Officer	10	18	0
6	104 Harry	1991-02-15	Assistant Central Intelligenc...	8	28	5
7	101 James	1990-12-15	Deputy Central Intelligence...	9	29	0

```

update case_assign set status='completed' date_of_comp=current_date where case_id=902 and agent_id=102;
select * from case_assign;

```

**Data Output**

case_id	head_id	agent_id	date_of_comp	status	deadline	date_of_assign
1	903	1001	103 [null]	undergoing	2020-04-10	2020-02-29
2	904	1001	104 2020-03-30	completed	2020-02-20	2019-01-20
3	902	1001	102 2020-03-30	completed	2020-03-01	2020-01-20
4	907	1002	107 2018-09-17	completed	2018-10-17	2018-07-17
5	908	1002	108 2018-07-16	completed	2018-09-16	2018-06-16
6	909	1002	109 2019-01-20	completed	2019-02-21	2019-01-10
7	905	1001	105 [null]	undergoing	2020-03-01	2020-01-10

```

select * from agent

```

**Data Output**

agent_id	name	dob	post	year_of_service	age	star
1	107 Charles	1983-10-10	Deputy Central Intelligence ...	16	36	0
2	110 Thomas	1969-09-01	Deputy Central Intelligence ...	18	50	5
3	103 Oliver	1975-02-01	Assistant Central Intelligenc...	15	44	10
4	111 kapil	2002-02-01	Junior Intelligence Officer	10	18	0
5	104 Harry	1991-02-01	Assistant Central Intelligenc...	8	28	5
6	102 Jake	1966-04-01	Deputy Central Intelligence ...	20	53	10
7	101 James	1990-12-01	Deputy Central Intelligence ...	9	29	0

2.	Fire_ag	Once an agent is fired, his id is place in old_ag relation and the cases are suspended	After delete	Old_ag,login_agent
----	---------	--	--------------	--------------------

QUERY:-

```
create trigger fire_ag
```

```
after delete on login_agent
```

for each row

execute procedure ag\_fire()

create or replace function ag\_fire()

returns trigger

as \$\$

begin

insert into old\_ag values(old.agent\_id);

update case\_assign set status='suspended' where agent\_id=old.agent\_id;

return new;

end;\$\$

language plpgsql

The screenshot shows the pgAdmin 4 interface with a database named 'TEAM\_REPORT of dhanush'. In the left sidebar, under 'Tables (24)', the 'ag\_post' table is selected. A query is being run in the 'Query Editor' tab:

```
1 select * from login_agent
```

The 'Data Output' tab displays the results of the query:

	agent_id	login_id	password
1	101	jameslia.in	jame
2	102	jakeslia.in	jake
3	103	oliverslia.in	oliver
4	104	harryslia.in	harry
5	105	charlieslia.in	charlie
6	107	jameselia.in	james
7	108	williamslia.in	william
8	109	oscarlia.in	oscar
9	110	thmasslia.in	thomas
10	111	agentkapil111	111kapil
11	106	mack	106mack

```

File Object Tools Help
Dashboard Properties SQL Statistics Dependencies Dependents Criminal/postgre... public.case_ass... Criminal/postgre...
Object Materialized Views Procedures Functions Sequences Tables Columns Constraints Indexes Rules Triggers
Criminal/postgres@PostgreSQL_12 Query History Query Editor
1 delete from login_agent where agent_id=106;
2 select * from old_ag;

Data Output Explain Messages
agent_id
1 [PK] numeric
1 106

```

```

File Object Tools Help
Dashboard Properties SQL Statistics Dependencies Dependents Criminal/postgre... public.case_ass... Criminal/postgre...
Object Materialized Views Procedures Functions Sequences Tables Columns Constraints Indexes Rules Triggers
Criminal/postgres@PostgreSQL_12 Query History Query Editor
1 select * from case_assign

Data Output Explain Messages
case_id head_id agent_id date_of_comp_date status
1 903 1001 103 [null] undergoing
2 904 1001 104 2020-03-30 completed
3 902 1001 102 2020-03-30 completed
4 906 1002 106 [null] suspended
5 907 1002 107 2018-09-17 completed
6 908 1002 108 2018-07-16 completed
7 909 1002 109 2019-01-20 completed
8 905 1001 105 [null] undergoing
9 901 1001 101 [null] undergoing
10 910 1002 109 2020-03-30 completed

```

3.	spec	Used to update the specialisation of an agent as soon as he/she completes a mission	After update	Case_assign,case_det, ag_skill
----	------	---	--------------	--------------------------------

QUERY:-

create trigger spec

after update on case\_assign

for each row

execute procedure spec\_ag()

create or replace function spec\_ag()

returns trigger

as \$\$

declare

deta cursor for select specialisation from (select specialisation,count(specialisation) from case\_assign  
natural join case\_det where agent\_id=old.agent\_id and status='completed'

group by specialisation)as temp3 natural join

(select max(count) as count from (select specialisation,count(specialisation) from case\_assign natural join  
case\_det where agent\_id=old.agent\_id and status='completed'

group by specialisation) as temp) as temp2;

v\_temp varchar;

begin

open deta;

if old.status='undergoing' and new.status='completed' then

delete from ag\_skill where agent\_id=old.agent\_id;

loop

fetch from deta into v\_temp;

exit when not found;

insert into ag\_skill (agent\_id,skill) values(old.agent\_id,v\_temp);

end loop;

end if;

```
close data;
```

```
return new;
```

```
end;$$
```

```
language plpgsql
```

The screenshot shows the pgAdmin 4 interface with a connection to 'Criminal/postgres@PostgreSQL 12'. The left sidebar displays the schema structure under 'login\_agent' and 'suspects' tables. The 'Query Editor' tab contains the following SQL query:

```
select * from ag_skill where agent_id=103
```

The 'Data Output' tab shows the results of the query:

agent_id	skill
103	catching thieves

The screenshot shows the pgAdmin 4 interface with a connection to 'Criminal/postgres@PostgreSQL 12'. The left sidebar displays the schema structure under 'login\_agent' and 'suspects' tables. The 'Query Editor' tab contains the following SQL queries:

```
update case_assign set date_of_comp=current_date,status='completed' where agent_id=103;
select * from ag_skill where agent_id=103;
```

The 'Data Output' tab shows the results of the second query:

agent_id	skill
103	investigating

4.	Log_in	Used to generate login and password for a new agent	Before insert	Agent,login_agent
----	--------	---	---------------	-------------------

QUERY:-

```

create trigger log_in
before insert on agent
for each row
execute procedure user_prod();

create or replace function user_prod()
returns trigger
as $$

declare
v_user varchar;v_pass varchar;

begin
v_user='agent'||''||new.name||''||new.agent_id;
v_pass=new.agent_id||''||new.name;
insert into login_agent (agent_id,login_id,password) values(new.agent_id,v_user,v_pass);
end;$$

language plpgsql

DO $$

<<first_block>>

declare

```

```

v_temp numeric;

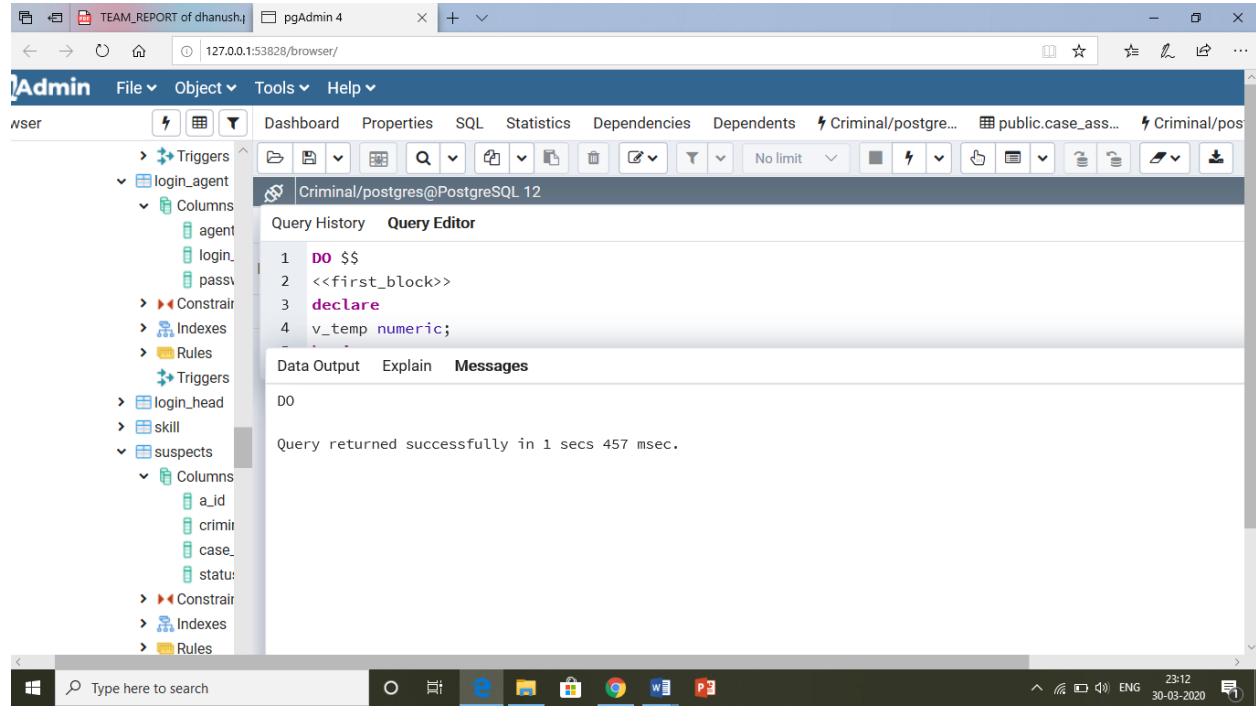
begin

select ag_id_gen() into v_temp;

insert into agent values(v_temp,'kapil','2002-02-20','Junior Intelligence Officer',10,18,0);

END first_block $$;

```



The screenshot shows the pgAdmin 4 interface with two windows. The top window displays a DO block that was executed successfully. The bottom window shows a query to select all from the login\_agent table, which returned 12 rows of data.

```

DO $$
<<first_block>>
declare
v_temp numeric;
begin

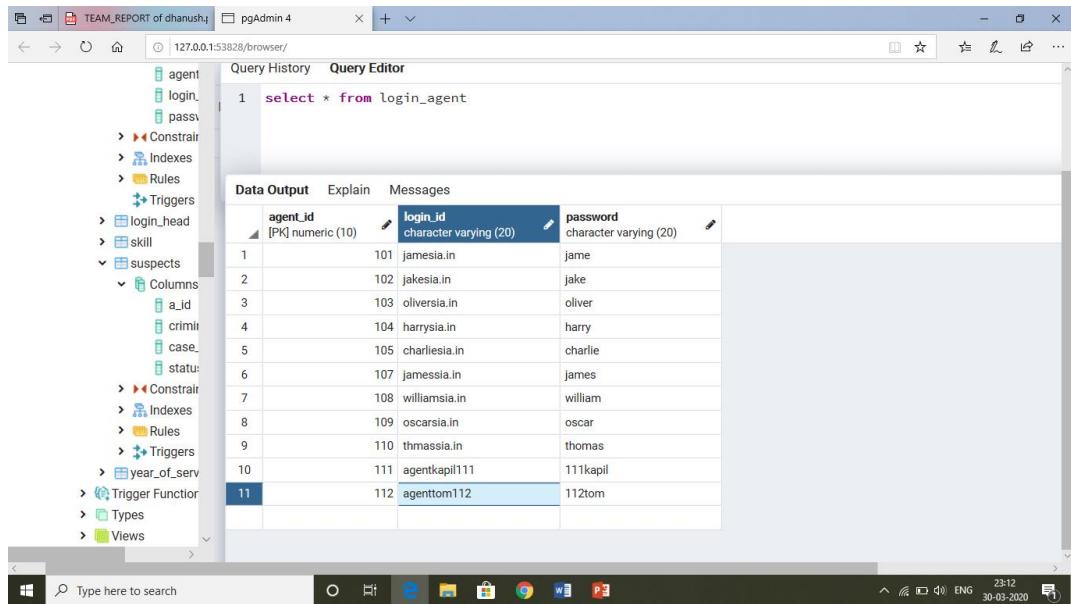
select ag_id_gen() into v_temp;

insert into agent values(v_temp,'kapil','2002-02-20','Junior Intelligence Officer',10,18,0);

END first_block $$;

```

agent_id	login_id	password
101	jamesia.in	jame
102	jakesia.in	jake
103	oliversia.in	oliver
104	harrysia.in	harry
105	charliesia.in	charlie
107	jamessia.in	james
108	williamsia.in	william
109	oscarzia.in	oscar
110	thmassia.in	thomas
111	agentkapil111	111kapil
112	agenttom112	112tom



The screenshot shows the pgAdmin 4 interface with two windows. The top window displays a DO block that was executed successfully. The bottom window shows a query to select all from the login\_agent table, which returned 12 rows of data.

```

select * from login_agent

```

agent_id	login_id	password
101	jamesia.in	jame
102	jakesia.in	jake
103	oliversia.in	oliver
104	harrysia.in	harry
105	charliesia.in	charlie
107	jamessia.in	james
108	williamsia.in	william
109	oscarzia.in	oscar
110	thmassia.in	thomas
111	agentkapil111	111kapil
112	agenttom112	112tom

## **FRONT END AND BACK END DETAILS**

For this project , our team has used

- HTML for structuring the webpage
- CSS for formatting the webpage
- Javascript for validation and image processing

in the Front End.

For back end , we have used PHP and have run the programs in localhost.

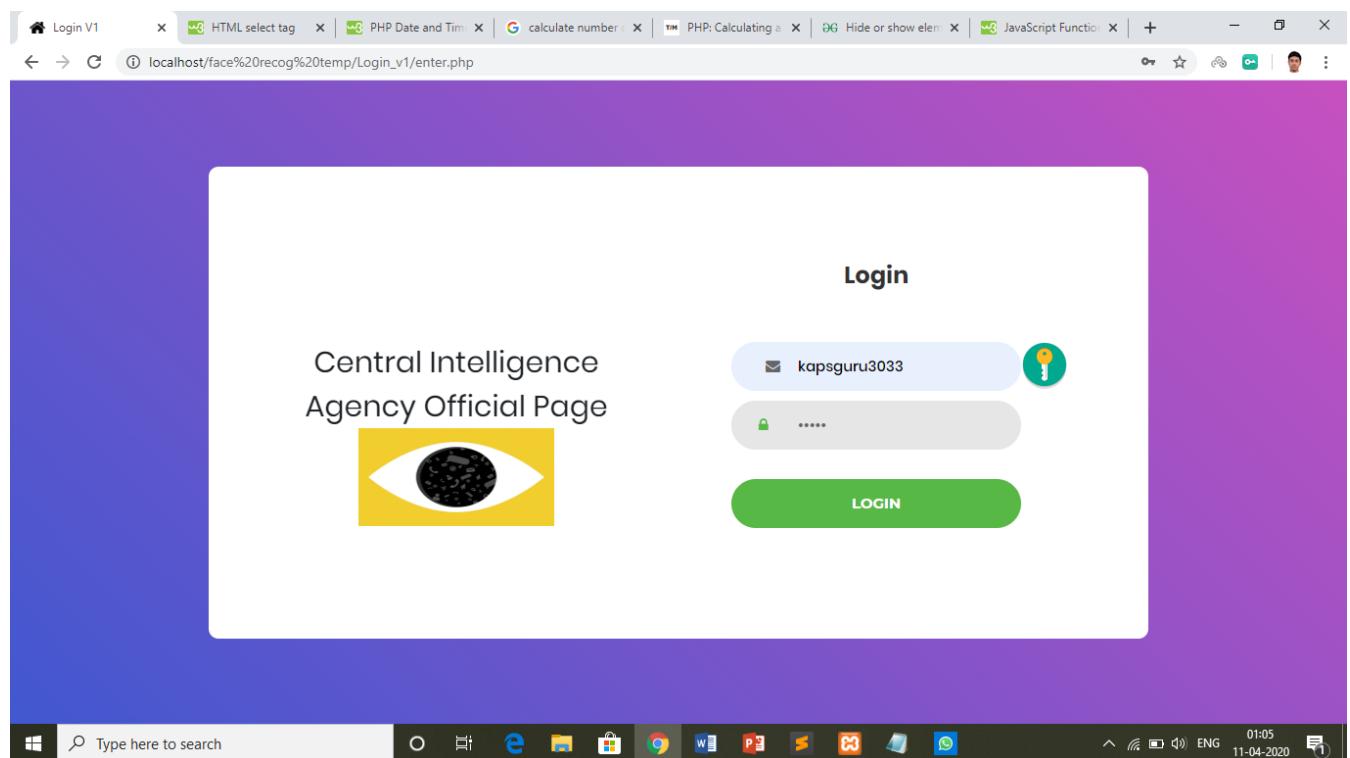
The Database that has been used is POSTGRES VERSION 12.

From the webpages that we have planned to do , 4 webpages has been submitted here.

# WEBPAGE 1

## LOGIN PAGE

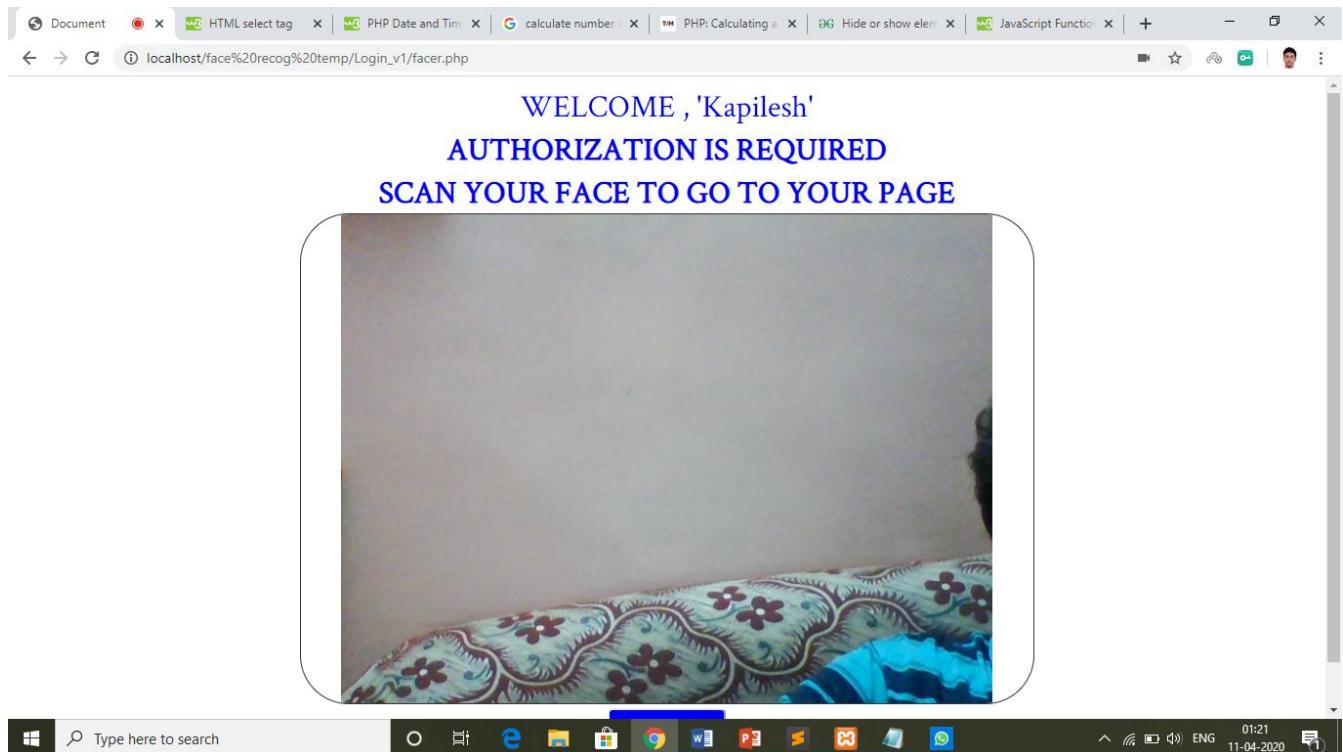
The agent or the head official is supposed to enter a valid username and password to go into their page.



# WEBPAGE 2

## FACE RECOGNITION

After entering the user id and password , the corresponding person's image is fetched from the database.Then , it is compared with the current image by capturing a photo through the device's camera. If both resembles , the main page is opened.



# WEBPAGE 3

## COMPLETED CASES

It lists all the completed cases of the agent . It is ordered in descending order of the cases which has happened long time ago.

The screenshot shows a web browser window with the URL [localhost/face%20recog%20temp/Login\\_v1/page2.php](http://localhost/face%20recog%20temp/Login_v1/page2.php). The page has a yellow header bar with a logo of an eye and the text "SURVEILLANCE INTELLIGENCE AGENCY". Below the header is a navigation bar with tabs: "CURRENT CASE" (disabled), "COMPLETED CASES" (selected), "PERSONAL DETAILS", and "MAKE A SEARCH". The main content area is titled "COMPLETED CASES" and contains a table with the following data:

S.NO.	CASE ID	CASE NAME	DAYS BEFORE WHICH THE CASE HAPPENED	
1	907	Robbery in bank	1352 days ago	<a href="#">view details</a>
2	903	Bombing	382 days ago	<a href="#">view details</a>
3	908	Murder of ex-minister	236 days ago	<a href="#">view details</a>
4	909	Escape of theif	193 days ago	<a href="#">view details</a>

# WEBPAGE 4

## REGISTERING AGENT DETAILS

The head official can enter a newly recruited agent by registering his details. Details like age,username and password for that agent are auto generated after registration.

SURVEILLANCE INTELLIGENCE AGENCY

CURRENT CASE    COMPLETED CASES    PERSONAL DETAILS    MAKE A SEARCH    ADD AN AGENT

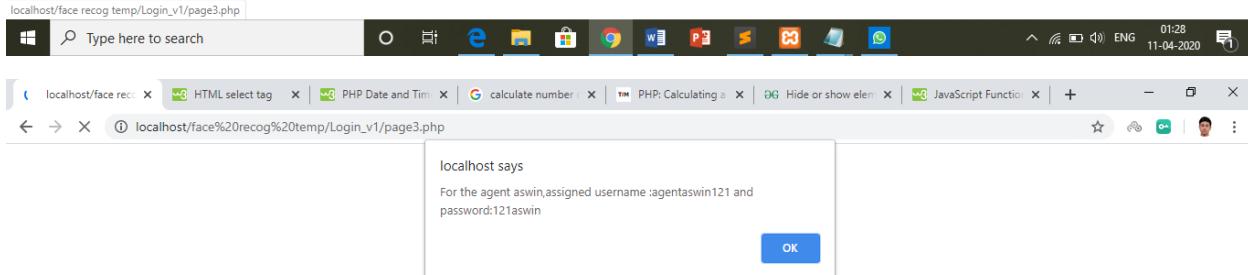
ENTER AGENT DETAILS

NAME

DOB  dd - mm - yyyy

POST  Deputy Central Intelligence Officer

Username, Password and Age of the agents are automatically generated



THANK YOU

