

DBMS Mini Project

---

# CORE (MINI)BANKING MANAGEMENT SYSTEM

---

Haneyah Seemein

Section: K

Semester: 5

## Description

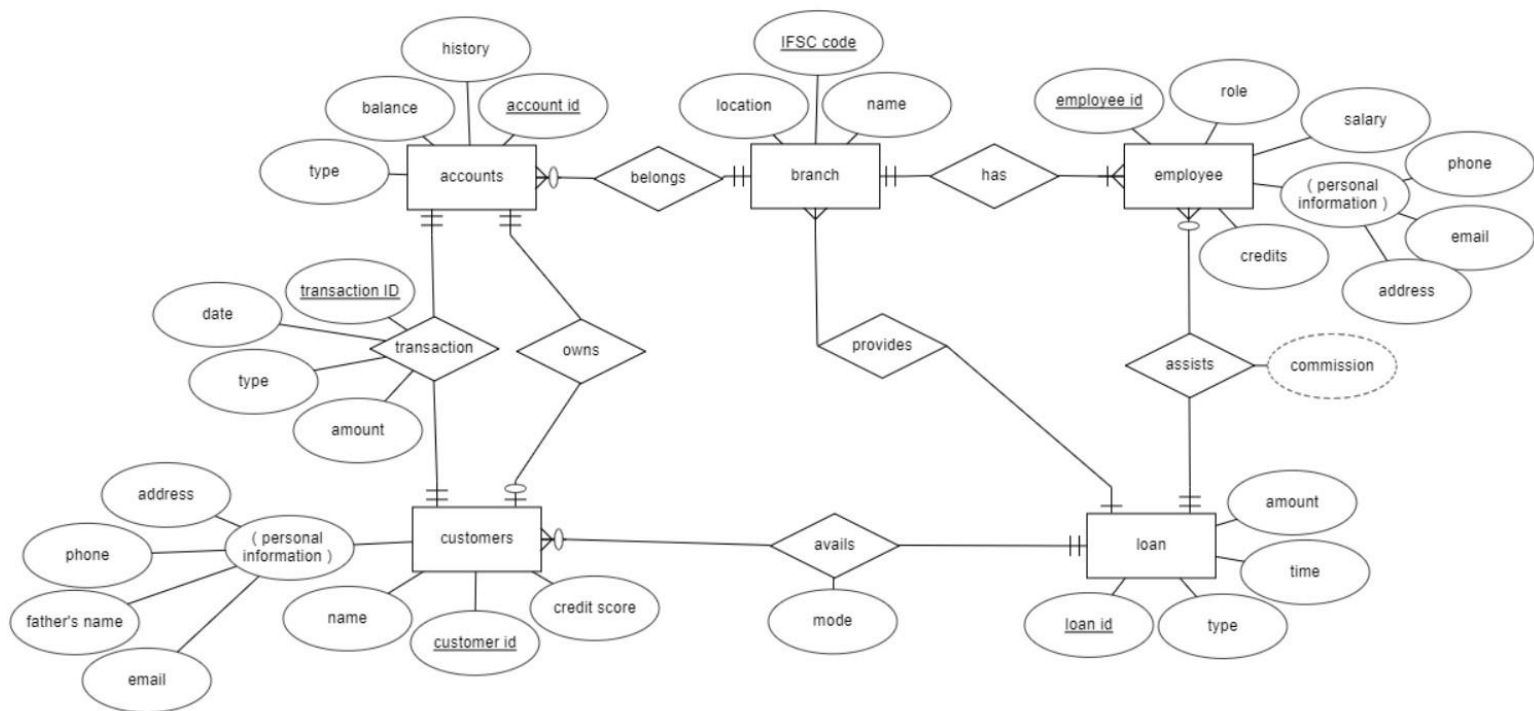
In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank. The Bank Account Management System undertaken as a project is based on relevant technologies.

This project has been developed to carry out the processes easily and quickly, which is not possible with the manual systems, which are overcome by this software. This project is developed using Python language and MYSQL use for database connection.

The system is designed as an interactive and content management system. The content management system deals with data entry while the interactive system deals with system interaction with the admin, employee and users. The features of this project will save transaction time and therefore increase the efficiency of the system. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view his profile and history of his account.

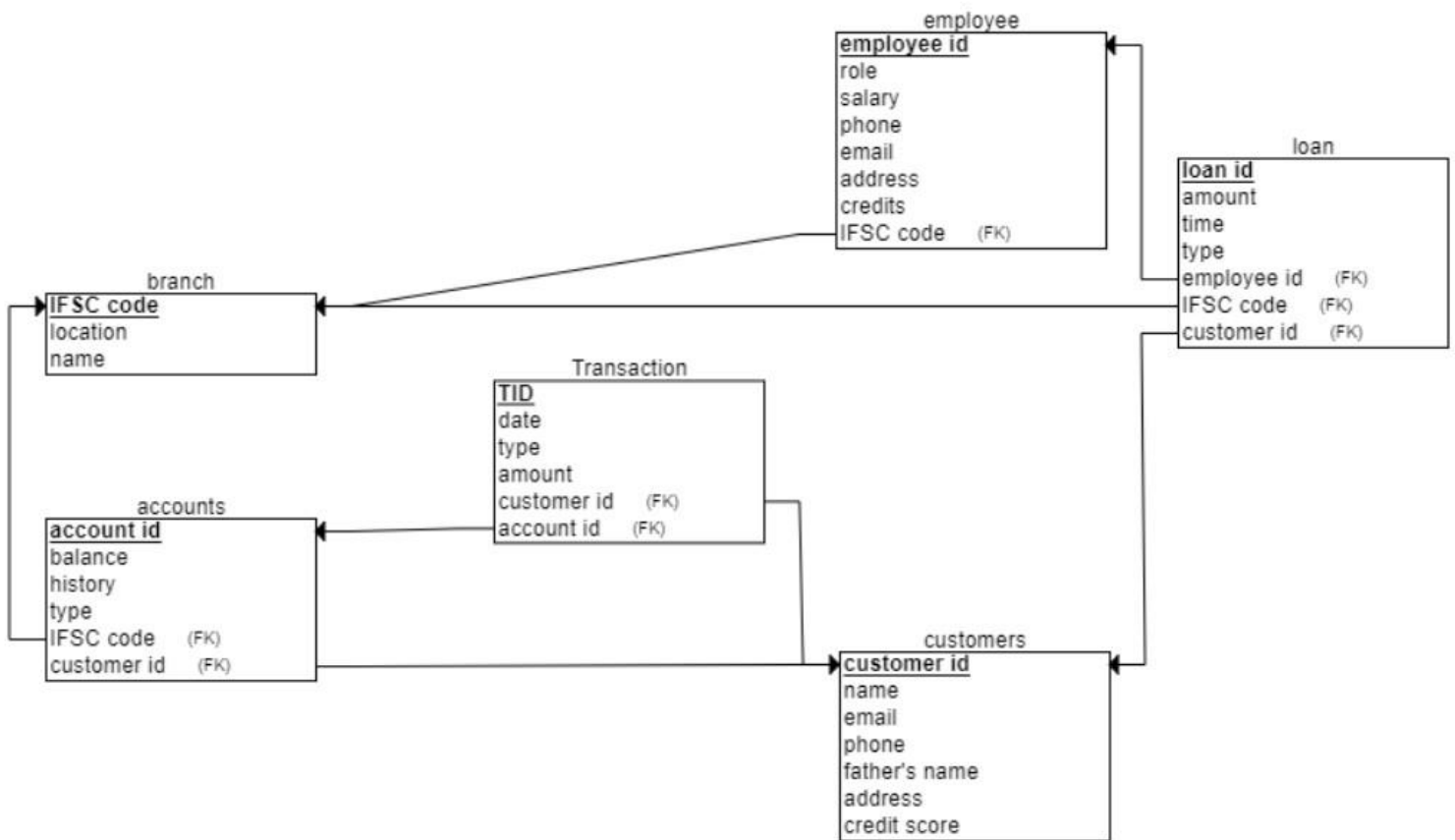
**This project has digitalized the basic functions of a bank. It enables the admin to have complete access over the bank. The employees are provided with options to create a customer's account, apply for a loan by a customer, make debit and credit transactions to a customer on request and also view his (loan) customer details. The customer can view his bank profile, account details, loan details and also the transaction history.**

## ER diagram



The given ER Diagram represents the superficial relationship between entities in a Bank. It is a simple outlook of a bank with basic required functionalities.

- **Account:** The account belongs to a customer of the bank and has a unique account\_id to identify each account. The balance and the transaction history of the same is also tracked by various functionalities.
- **Branch:** Each branch/unit contains the employees, accounts and an IFSC code.
- **Employee:** The employees perform various operations to assist the customer. The employees help the customers for availing loans.
- **Customer:** The customer with his profile containing his name, personal information and credit score may or may not have an account and loan.
- **Loan:** created by the employee for the customer, has a unique identifier loan\_id, time period, amount and the loan type like personal, home, education etc.



## Relational Schema

## Building Database

### Creating branch

```
CREATE TABLE branch
(
  name VARCHAR(255) NOT NULL,  IFSC INT(4) NOT NULL,  location VARCHAR(255) NOT
  NULL,
  PRIMARY KEY (IFSC)
);
```

### Creating customer

```
CREATE TABLE customer
(
  customer_id INT(7) NOT NULL,  name VARCHAR(255) NOT NULL,  phone VARCHAR(10) NOT
  NULL,  email VARCHAR(255) NOT NULL,  address VARCHAR(255) NOT NULL,  credit_score
  INT(1) NOT NULL,
  PRIMARY KEY (customer_id)
);
```

## Creating employee

```
CREATE TABLE employee
(
    employee_id INT(7) NOT NULL,
    name VARCHAR(255) NOT NULL,
    role VARCHAR(10) NOT NULL,
    salary INT NOT NULL,    phone
    VARCHAR(10) NOT NULL,    email
    VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL,
    credits INT(1) NOT NULL,    IFSC
    INT(4) NOT NULL,
    PRIMARY KEY (employee_id),
    FOREIGN KEY (IFSC) REFERENCES branch(IFSC)
);
```

## Creating loan

```
CREATE TABLE loan
(
    loan_id INT(10) NOT NULL,
    amount INT NOT NULL,    time
    INT NOT NULL,    type
    VARCHAR(10) NOT NULL,
    customer_id INT(7) NOT NULL,
    employee_id INT(7) NOT NULL,
    IFSC INT(4) NOT NULL,
    PRIMARY KEY (loan_id),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
    FOREIGN KEY (employee_id) REFERENCES employee(employee_id),
    FOREIGN KEY (IFSC) REFERENCES branch(IFSC)
);
```

## Creating account

```
CREATE TABLE account
(
    account_id INT(10) NOT
    NULL,    balance INT NOT NULL,
    type VARCHAR(10) NOT NULL,
    IFSC INT(4) NOT NULL,
    customer_id INT(7) NOT NULL,
    PRIMARY KEY (account_id),
    FOREIGN KEY (IFSC) REFERENCES branch(IFSC),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);
```

## Creating transaction

```
CREATE TABLE Transaction
(
  TID INT(10) NOT NULL,
  date INT(8) NOT NULL,
  amount INT NOT NULL,
  account_id INT(10) NOT NULL,
  customer_id INT(7) NOT NULL,
  PRIMARY KEY (TID),
  FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
  FOREIGN KEY (account_id) REFERENCES account(account_id)
);
```

## Populating the database

### Inserting into branch

```
INSERT INTO branch values ('Bangalore', 1000, "inside Mantri Square");
INSERT INTO branch values ('Bangalore', 2000, "Banashankari");
INSERT INTO branch values ('Bombay', 3000, "Juhu Beach");
INSERT INTO branch values ('Delhi', 4000, "Lajpath Nagar");
INSERT INTO branch values ('Kolkatta', 5000, "RT Port");
```

### Inserting into customer

```

insert into customer values (64852, 'Merrily Pursehouse',
'5093920400',
'mpursehousep@jigsy.com', '70 Southoff Center', 10); insert into
customer values (66505, 'Winny Cribbin', '4283483313',
'wcribbing@1688.com', '442 Manufacturers Court', 1); insert into
customer values (64393, 'Jackie Deehan', '1716937276',
'jdeehanr@ca.gov', '47 Ronald Regan Point', 1); insert into customer
values (51221, 'Munroe Stilliard', '1674746391',
'mstilliards@tuttocitta.it', '5601 Lunder Point', 4); insert into
customer values (23644, 'Charis Vampouille', '4288211430',
'cvampouillet@ning.com', '4 Buhler Court', 2); insert into
customer values (94947, 'Angele Chesman', '3049219959',
'achesmanu@illinois.edu', '85424 Iowa Pass', 8); insert into
customer values (89135, 'Justinian Houson', '1298349681',
'jhousonv@theguardian.com', '480 Fairview Center', 5); insert into
customer values (52613, 'Titos Slemmonds', '1261738108',
'tslemmondsw@imageshack.us', '1 Coolidge Way', 3); insert into
customer values (63181, 'Koren Franckton', '5666689493',
'kfrancktonx@instagram.com', '29401 Mayfield Avenue', 3);
insert into customer values (61121, 'Sherie Mapam', '3145586035',
'smapamy@reuters.com', '40 Gina Point', 3); insert into
customer values (57692, 'Jena Dabel', '2832115413',
'jdabelz@yellowbook.com', '2 Rieder Point', 7); insert into
customer values (59137, 'Carley Jenoure', '6103297150',
'cjenoure10@mapquest.com', '039 Forest Run Crossing', 4);
insert into customer values (85846, 'Manny Fullun', '3049730077',
'mfullun11@sogou.com', '8944 Butterfield Avenue', 9); insert into
customer values (69516, 'Alyssa Roxburch', '7122461168',
'aroxburch12@wiley.com', '404 Eagle Crest Point', 8);
insert into customer values (79457, 'Quentin Stoter', '5716892932',
'qstoter13@foxnews.com', '6 Arizona Hill', 9); insert into customer
values (88958, 'Buck Simoncello', '9943061255',
'bsimoncello14@163.com', '68271 Oak Road', 3); insert into customer
values (62986, 'Koressa Parlor', '7931091434',
'kparlor15@cam.ac.uk', '516 Paget Hill', 10);

```



```
insert into customer values      ,  
                                (67691 'Cheri Winsen', '3594482283',  
'cwinser16@wufoo.com', '9521 Algoma Road', 1);
```

```

insert into customer values      ,
insert into customer values (69379, 'Ginnie Khomin', '4279423817',
'gkhomin17@nbcnews.com', '462 Golf Course Park', 8); insert into
customer values (84510, 'Terrance Scrivin', '8225350096',
'tscrivin18@paginegialle.it', '46 Cottonwood Terrace', 3);
insert into customer values (87916, 'Barney Hearst', '9761576430',
'bhearst19@webeden.co.uk', '717 Eagan Crossing', 2); insert into
customer values (96302, 'Gawain Purrier', '3547511059',
'gpurrier1a@salon.com', '7 Fair Oaks Hill', 5); insert into
customer values (16113, 'Leoine Dudgeon', '7168752183',
'ldudgeon1b@people.com.cn', '6 Evergreen Point', 8); insert into
customer values (72869, 'Genna Sauvain', '4692464694',
'gsauvain1c@nps.gov', '8568 Springs Avenue', 5); insert into
customer values (70533, 'Katie Tersay', '6324603028',
'ktersay1d@wunderground.com', '37350 Shoshone Point', 1);
insert into customer values (67881, 'Erasmus Connar', '5761536154',
'econnar1e@reference.com', '22 Gale Avenue', 3); insert into
customer values (02952, 'Elizabeth Farey', '9368802991',
'efarey1f@newyorker.com', '0449 Comanche Lane', 6); insert into
customer values (97370, 'Tanya Novakovic', '7358484674',
'tnovakovic1g@newyorker.com', '96910 Northport Trail', 6);
insert into customer values (04120, 'Serena Manginot', '9454940689',
'smanginot1h@multiply.com', '4 Sycamore Terrace', 2); insert
into customer values (67912, 'Bald Coopey', '5851956684',
'bcoopey1i@zimbio.com', '28 Mitchell Hill', 9); insert into
customer values (73174, 'Galvan Shurman', '7449290604',
'gshurman1j@nydailynews.com', '16 Express Street', 1); insert into
customer values (81871, 'Roberto Roelofsen', '6788461039',
'rroelofsen1k@wordpress.com', '059 Texas Alley', 6); insert into
customer values (52395, 'Flore Brazier', '9792538962',
'fbrazier1l@tiny.cc', '6512 West Pass', 9);
insert into customer values (03141, 'Hilary Welband', '3977766598',
'hwelband1m@msu.edu', '44 Knutson Pass', 6);
insert into customer values (41662, 'Monroe Lightoller', '3529053051',
'mlightoller1n@pcworld.com', '3385 Waubesa Place', 9);
insert into customer values (66352, 'Almeta Pegg', '3862054681',
'apegg1o@cnbc.com', '5978 Banding Drive', 6); insert into customer
values (56241, 'Maryellen Edridge', '5758256168',
'medridge1p@godaddy.com', '4925 Blaine Junction', 6);
insert into customer values (81321, 'Nicolais Aaronson', '6013237728',
'naaronson1q@printfriendly.com', '91 Reinke Center', 3);
insert into customer values (98837, 'Josiah Reck', '1874127093',
'jreck1r@amazonaws.com', '4836 Pierstorff Center', 4); insert into
customer values (31298, 'Analise Downse', '5187114870',
'adownse1s@aol.com', '26735 Orin Center', 1); insert into
customer values (91000, 'Elka Cassey', '1585640565',
'ecassey1t@about.com', '482 David Avenue', 1);

```

```
insert into customer values      ,  
                                (22874 'Kendrick de Clerk', '6313086001',  
'kde1u@fema.gov', '67675 Mariners Cove Lane', 1);
```

```

insert into customer values      ,
insert into customer values (02642, 'Sophronia Postins', '1164260229',
'spostins1v@zimbio.com', '62 Nobel Hill', 4); insert into customer
values (21858, 'Ethelind Lardeur', '7787323996',
'elardeur1w@tumblr.com', '74 Carioca Center', 2); insert into
customer values (97408, 'Darcee Radbone', '4882238562',
'dradbone1x@mapquest.com', '29499 Tennessee Avenue', 9); insert
into customer values (06472, 'Deeyn Rappport', '4953430073',
'drappport1y@who.int', '730 Emmet Way', 2); insert into customer
values (55105, 'Wandie Tobin', '1793788178',
'wtobin1z@microsoft.com', '2638 Butternut Hill', 5); insert into
customer values (12174, 'Shawna Corbin', '1199864941',
'scorbin20@nba.com', '4 Claremont Parkway', 6); insert into
customer values (49971, 'Lucius Tures', '8619398073',
'ltures21@wisc.edu', '3527 Tennyson Point', 5); insert into
customer values (87377, 'Colby Coate', '9847297283',
'ccoate22@sohu.com', '98 Anzinger Plaza', 2); insert into
customer values (61785, 'Frances Sirey', '1999055760',
'fsirey23@google.ru', '34019 Glendale Parkway', 9); insert into
customer values (87565, 'Ange Streatley', '2798403804',
'astreatley24@weebly.com', '3003 Monterey Lane', 6); insert into
customer values (54513, 'Baron Schimmang', '9803060709',
'bschimmang25@answers.com', '31861 Starling Hill', 4);
insert into customer values (62773, 'Eden Vink', '6376902067',
'evink26@google.co.jp', '58 Macpherson Lane', 7); insert into
customer values (27172, 'Dulcea Mariault', '7716922129',
'dmariault27@cafepress.com', '67 Mosinee Court', 2); insert into
customer values (39918, 'Adella Pegden', '2434701952',
'apegden28@bloomberg.com', '35 Bultman Place', 10); insert into
customer values (92203, 'Gilbert Wibrew', '6889471547',
'gwibrew29@time.com', '63 Bashford Pass', 2);
insert into customer values (57389, 'Adolph Maciejak', '8131414736',
'amaciejak2a@marketwatch.com', '67 Mitchell Court', 1);
insert into customer values (86763, 'Heindrick Helversen', '3157535230',
'hhelversen2b@theglobeandmail.com', '3080 Dottie Point', 9);
insert into customer values (07632, 'Sheela Hundell',
'2157644448',
'shundell2c@desdev.cn', '095 Buell Place', 2); insert into
customer values (92362, 'Dael Ovendale', '7006326975',
'dovendale2d@jugem.jp', '5911 Saint Paul Lane', 5);
insert into customer values (70047, 'Benedicto Jeeks', '7043476534',
'bjeeks2e@nifty.com', '9 Lyons Circle', 7); insert into
customer values (91859, 'Agosto Pahl', '5892227603',
'apahl2f@hhs.gov', '39306 Northridge Pass', 3); insert into
customer values (63704, 'Chrotoem Jaouen', '8298739546',
'cjaouen2g@state.gov', '2977 Northland Avenue', 8); insert into
customer values (32342, 'Daren Grube', '8561538609',
'dgrube2h@jugem.jp', '9 Sycamore Junction', 7);

```

```

insert into customer values
    (95794, 'Carlynn Riches', '6271346985',
    'criches2i@alexa.com', '794 Farragut Junction', 1);
insert into customer values (51164, 'Margy Iron', '7691471618',
'miron2j@prweb.com', '015 Clemons Point', 4); insert into customer
values (58546, 'Sinclair Jonson', '4295393742',
'sjonson2k@nydailynews.com', '72530 Corben Junction', 8); insert
into customer values (28226, 'Janka Kulic', '2182858611',
'jkulic2l@skype.com', '0142 Logan Court', 10); insert into
customer values (63202, 'Chrissy Checo', '3168441646',
'ccheco2m@salon.com', '9803 Prairieview Plaza', 5); insert into
customer values (78658, 'Buckie De Michetti', '3552222548',
'bde2n@zdnnet.com', '9 Lakeland Park', 5); insert into customer
values (63514, 'Mallorie Parvin', '4954180279',
'mparvin2o@comsenz.com', '6 Shasta Terrace', 9); insert into
customer values (59874, 'Amandie Fairhead', '6541638614',
'afairhead2p@wp.com', '512 Stephen Parkway', 10); insert into customer
values (09382, 'Juliette Nickerson', '2996923401',
'jnickerson2q@csmonitor.com', '9 Prairieview Crossing', 7);
insert into customer values (29347, 'Austine Lyptrit', '8408534229',
'alyptrit2r@businessweek.com', '20 Fisk Circle', 6);

```

## Inserting into employees

```

insert into employee (employee_id, name, role, salary, phone, email,
address, credits, IFSC) values ('000', 'DUMMY', 'DUMMY',
1000000, '9999999999',
'richierich@google.com.au', 'DUMMY', 0, 5000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('320', 'Agretha Embra', 'Advisor', 4000, '4967270634',
'aembra0@google.com.au', '44 Northview Terrace', 9, 5000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('190', 'Bria Coslitt', 'Advisor', 4000, '8677149517',
'bcoslitt1@blinklist.com', '9714 Green Ridge Court', 8, 2000);
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('924', 'Ira Comusso', 'Advisor', 4000, '7616891181',
'icomusso2@soup.io', '9 Farragut Circle', 2, 5000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('189', 'Ragnar Brayson', 'Advisor', 4000, '1703786970',
'rbrayson3@networksolutions.com', '1 Vera Hill', 5, 5000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('416', 'Jereme Castenda', 'Advisor', 4000, '4952204341',
'jcastenda4@plala.or.jp', '3933 Menomonie Way', 6, 4000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('318', 'Lisle Gull', 'Advisor', 4000, '9068674982',
'lgull15@nih.gov', '4769 Homewood Court', 6, 4000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('503', 'Angie Menco', 'Advisor', 4000, '2747264393',

```

```
insert into customer values  
'amenco6@lycos.com', '7116 Butterfield Alley', 3, 3000);
```

```

insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('241', 'Cris Bazley', 'Advisor', 4000, '3429789990',
'cbazley7@cbc.ca', '40072 Linden Place', 1, 2000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('871', 'Augusto Lafranchi', 'Advisor', 4000, '4563642404',
'alafranchi8@blinklist.com', '2788 Linden Lane', 4, 5000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('430', 'Ade Mathonnet', 'Advisor', 4000, '1843620257',
'amathonnet9@hugedomains.com', '6891 Dexter Road', 1, 2000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('558', 'Tammie Grogan', 'Advisor', 4000, '4269715539',
'tgrogana@4shared.com', '1122 Annamark Road', 5, 2000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('352', 'Latashia Berry', 'Advisor', 4000, '4243885176',
'lberryb@wordpress.com', '58078 Spenser Center', 1, 3000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('433', 'Peyter Parsisson', 'Advisor', 4000,
'4196445206', 'pparsissonc@hugedomains.com', '0406 Hermina Street', 5, 3000);
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('957', 'Jojo Stiggers', 'Officer', 17000, '4147849108',
'jstiggersd@ucsd.edu', '0419 Heffernan Plaza', 3, 5000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('288', 'Corrie Samworth', 'Officer', 17000, '9023834850',
'csamworthe@dedecms.com', '335 Warner Parkway', 10, 5000); insert into
employee (employee_id, name, role, salary, phone, email, address, credits,
IFSC) values ('011', 'Hazel Jills', 'Officer', 17000, '7526358691',
'hjillsf@ifeng.com', '7 Cascade Court', 2, 5000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('139', 'Renaldo Heake', 'Officer', 17000, '2057143941',
'rheakeg@nature.com', '570 Holmberg Avenue', 6, 5000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('029', 'Ninnette Normington', 'Officer', 17000, '3229525216',
'nnormingtonh@foxnews.com', '4 Oriole Plaza', 3, 3000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('950', 'Nananne Verney', 'Officer', 17000, '1998019254',
'nverneyi@ox.ac.uk', '4 Paget Pass', 3, 3000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('497', 'Bronnie Hearle', 'Officer', 17000, '6304110149',
'bhearlej@free.fr', '9254 Mallory Way', 6, 3000); insert into employee
(employee_id, name, role, salary, phone, email, address, credits, IFSC)
values ('600', 'Shelley Banthorpe', 'Officer', 17000,
'3175913523', 'sbanthorpek@networkadvertising.org', '68772 South Trail', 1,
2000); insert into employee (employee_id, name, role, salary, phone, email,
address, credits, IFSC) values ('187', 'Marwin Duffin', 'Officer', 17000,
'2005678910',
'mduffinl@reverbnation.com', '92 Transport Terrace', 9, 3000);

```

```
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('689', 'Katey Korpola', 'Manager', 160000,
'6336154354', 'kkorpola@rakuten.co.jp', '5257 Little Fleur Pass', 2, 2000);
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('199', 'Kerrill Happer', 'Manager', 160000,
'1437745561', 'khappern@wikipedia.org', '76448 Becker Point', 6, 4000);
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('984', 'Cecilla Wooland', 'Manager', 160000,
'3093597506', 'cwoolando@slashdot.org', '04898 Golden Leaf Parkway', 4,
4000); insert into employee (employee_id, name, role, salary, phone, email,
address, credits, IFSC) values ('438', 'Myrta Leavesley', 'Branch Head',
70000, '2268435064', 'mleavesleyp@mapy.cz', '69834 Ryan Street', 3, 5000);
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('469', 'Cathrin Kenwyn', 'Branch Head', 70000,
'8079323282', 'ckenwynq@angelfire.com', '70 Everett Place', 10, 1000); insert
into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('692', 'Job Ingerith', 'Branch Head', 70000,
'6699526697', 'jingerithr@apple.com', '291 McCormick Center', 5, 2000);
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('472', 'Tallia Beacroft', 'Branch Head', 70000,
'4081853178', 'tbeacrofts@domainmarket.com', '8543 Bartelt Hill', 4, 3000);
insert into employee (employee_id, name, role, salary, phone, email, address,
credits, IFSC) values ('491', 'Britteny Rimell', 'Branch Head', 70000,
'8696856359', 'brimellt@vistaprint.com', '75603 Birchwood Avenue', 3, 4000);
```



## Join Queries

### 1. Customer's bank balance and the amount he/she owes to bank for loan.

```
SELECT customer.name, account.balance, loan.amount FROM customer INNER JOIN account
INNER JOIN loan where customer.customer_id=account.customer_id AND
loan.customer_id=customer.customer_id;
```

```
MariaDB [bank]> SELECT customer.name, account.balance, loan.amount FROM customer INNER JOIN account INNER JOIN loan where
customer.customer_id=account.customer_id AND loan.customer_id=customer.customer_id;
+-----+-----+-----+
| name      | balance | amount |
+-----+-----+-----+
| Hilary Welband | 200000 | 380000 |
| Haneyah Seemein | 3100 | 140000 |
+-----+-----+-----+
2 rows in set (0.015 sec)
```

### 2. Names of all the customers along with their addresses and account id who belong to the branch with IFSC code 2000.

```
SELECT branch.name, customer.name, account.account_id, branch.location,
customer.address FROM branch INNER JOIN account INNER JOIN customer WHERE
branch.IFSC=2000 AND customer.customer_id=account.customer_id;
```

```
MariaDB [bank]> SELECT branch.name, customer.name, account.account_id, branch.location, customer.address
-> FROM branch INNER JOIN account INNER JOIN customer
-> WHERE branch.IFSC=2000 AND customer.customer_id=account.customer_id;
+-----+-----+-----+-----+-----+
| name      | name      | account_id | location  | address      |
+-----+-----+-----+-----+-----+
| Bangalore | Haneyah Seemein | 2000 | Banashankari | Banashankari |
| Bangalore | Hilary Welband | 1111 | Banashankari | 44 Knutson Pass |
| Bangalore | Juliette Nickerson | 2020 | Banashankari | 9 Prairieview Crossing |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

### 3. All the employees working in the branches located in Bangalore

```
SELECT employee.name, employee.role, branch.location, employee.salary,
employee.credits FROM employee INNER JOIN branch WHERE branch.name='Bangalore' AND
employee.IFSC=branch.IFSC;
```

```
MariaDB [bank]> SELECT employee.name, employee.role, branch.location, employee.salary, employee.credits FROM employee INNER JOIN
branch WHERE branch.name='Bangalore' AND employee.IFSC=branch.IFSC;
```

name	role	location	salary	credits
Cathrin Kenwyn	Branch Hea	inside Mantri Square	70000	10
Bria Coslitt	Advisor	Banashankari	4000	8
Cris Bazley	Advisor	Banashankari	4000	1
Ade Mathonnet	Advisor	Banashankari	4000	1
Tammie Grogan	Advisor	Banashankari	4000	5
Katey Korpola	Manager	Banashankari	160000	2
Job Ingerith	Branch Hea	Banashankari	70000	5

```
7 rows in set (0.004 sec)
```

#### 4. Customers who made transactions of more than 1000 rupees

```
SELECT customer.name, transaction.TID, transaction.amount FROM customer JOIN
account JOIN transaction WHERE customer.customer_id=account.customer_id AND
transaction.account_id=account.account_id AND transaction.amount>1000;
```

```
MariaDB [bank]> SELECT customer.name, transaction.TID, transaction.amount FROM customer JOIN account JOIN transaction WHERE
customer.customer_id=account.customer_id AND transaction.account_id=account.account_id AND transaction.amount>1000;
```

name	TID	amount
Haneyah Seemein	44505	1400
Haneyah Seemein	47431	1400

```
2 rows in set (0.001 sec)
```

## Aggregate Functions

### 1. Maximum salary an employee gets in the bank

```
SELECT MAX(salary) FROM employee;
```

```
MariaDB [bank]> SELECT MAX(salary) FROM employee;
+-----+
| MAX(salary) |
+-----+
|      160000 |
+-----+
1 row in set (0.000 sec)
```

### 2. Average credits/ratings of officer.

```
SELECT AVG(credits) FROM employee WHERE role='Officer';
```

```
MariaDB [bank]> SELECT AVG(credits) FROM employee WHERE role='Officer';
+-----+
| AVG(credits) |
+-----+
|         5.2500 |
+-----+
1 row in set (0.001 sec)
```

### 3. Number of employees working for Bombay branch.

```
SELECT count(*) AS BombayEmployees FROM employee where IFSC=3000;
```

```
MariaDB [bank]> SELECT count(*) AS BombayEmployees FROM employee where IFSC=3000;
+-----+
| BombayEmployees |
+-----+
|                8 |
+-----+
1 row in set (0.001 sec)
```

### 4. Number of Advisors working for the bank

```
SELECT count(*) AS BranchHeads FROM employee where role='Advisor';
```

```
MariaDB [bank]> SELECT count(*) AS BranchHeads FROM employee where role='Advisor';  
+-----+  
| BranchHeads |  
+-----+  
|          13 |  
+-----+  
1 row in set (0.000 sec)
```

## Set Operations

1. IDs of customers whose loan amount is greater than 10000 and bank balance more than 2000

```
SELECT customer_id FROM loan WHERE amount>10000 INTERSECT SELECT
customer_id FROM account WHERE balance>2000;
```

```
MariaDB [bank]> SELECT customer_id FROM loan WHERE amount>10000 INTERSECT SELECT customer_id FROM account WHERE balance>2000;
+-----+
| customer_id |
+-----+
|          3141 |
|          2000 |
+-----+
2 rows in set (0.016 sec)
```

2. Names of the employees working for Bombay Branch along with Managers

```
SELECT name FROM employee WHERE IFSC=3000 UNION SELECT name FROM employee
WHERE role='Manager';
```

```
MariaDB [bank]> SELECT name FROM employee WHERE IFSC=3000 UNION SELECT name FROM employee WHERE role='Manager';
+-----+
| name |
+-----+
| Ninnette Normington |
| Marwin Duffin |
| Latashia Berry |
| Peyter Parsisson |
| Tallia Beacroft |
| Bronnie Hearle |
| Angie Menco |
| Nananne Verney |
| Kerrill Happer |
| Katey Korpola |
| Cecilla Wooland |
+-----+
11 rows in set (0.013 sec)
```

3. All the employees in Bangalore except the managers

```
SELECT name FROM employee WHERE IFSC=1000 EXCEPT SELECT name FROM employee
WHERE role='Manager';
```

```
MariaDB [bank]> SELECT name FROM employee WHERE IFSC=2000 EXCEPT SELECT name FROM employee WHERE role='Manager';
+-----+
| name |
+-----+
| Bria Coslitt |
| Cris Bazley |
| Ade Mathonnet |
| Tammie Grogan |
| Job Ingerith |
+-----+
5 rows in set (0.001 sec)
```

#### 4. Names of all good employees and customers for bonus/gifts

```
SELECT name FROM employee WHERE credits>8 UNION SELECT name FROM customer
WHERE credit_score>8;
```

```
MariaDB [bank]> SELECT name FROM employee WHERE credits>8 UNION SELECT name FROM customer WHERE credit_score>8;
+-----+
| name |
+-----+
| Marwin Duffin |
| Corrie Samworth |
| Agretha Embra |
| Cathrin Kenwyn |
| DUMMY |
| Haneyah Seemein |
| Janka Kulic |
| Adella Pegden |
| Monroe Lightoller |
| Flore Brazier |
| Amandie Fairhead |
| Frances Sirey |
| Koressa Parlor |
| Mallorie Parvin |
| Merrily Pursehouse |
| Bald Coopey |
| Quentin Stoter |
| Manny Fullun |
| Heindrick Helversen |
| Darcee Radbone |
+-----+
20 rows in set (0.001 sec)
```

## Function

```
DELIMITER $$
CREATE FUNCTION totalamount(amount INT, type VARCHAR(10))
RETURNS INT
BEGIN
    DECLARE interestpercent INT;
```

```

IF type = 'Home' THEN
    SET interestpercent = 5;
ELSEIF type = 'Personal' THEN
    SET interestpercent = 10;
ELSEIF type = 'Education' THEN
    SET interestpercent = 2;
ELSEIF type = 'Fund' THEN
    SET interestpercent = 0;
END IF;
RETURN amount * interestpercent + amount;
END $$
DELIMITER;

```

```

MariaDB [bank]> SELECT totalamount(20000,"Personal");
+-----+
| totalamount(20000,"Personal") |
+-----+
|                               |
|                               |
|                               |
+-----+
1 row in set (0.026 sec)

```

## View

```

CREATE VIEW [leads] AS SELECT
loan.loan_id,customer.name,loan.type,customer.phone,loan.amount FROM loan INNER
JOIN customer INNER JOIN employee
ON loan.customer_id = customer.customer_id AND loan.employee_id
=employee.employee_id

```

```

MariaDB [bank]> SELECT * FROM leads;
+-----+-----+-----+-----+-----+
| loan_id | name          | type   | phone    | amount |
+-----+-----+-----+-----+-----+
| 22112512 | Sophronia Postins | Home   | 1164260229 | 380000 |
| 211251209 | Hilary Welband    | Home   | 3977766598 | 380000 |
| 211281154 | Shawna Corbin     | Home   | 1199864941 | 200000 |
| 211281205 | Elizabeth Farey   | Personal | 9368802991 | 300000 |
| 2147483647 | Haneyah Seemein   | Fund   | 9108630164 | 140000 |
+-----+-----+-----+-----+-----+
5 rows in set (0.008 sec)

```

## Trigger

```

CREATE TRIGGER update_bal
  AFTER INSERT
  ON Transaction for each row BEGIN  update account set
balance=balance+new.amount where account.account_id=new.account_id;
END $$

```

Before:



```
MariaDB [bank]> SELECT * FROM account where account_id=2020;
```

account_id	balance	type	IFSC	customer_id
2020	20000	Savings	2000	9382

```
1 row in set (0.001 sec)
```

After:

```
MariaDB [bank]> INSERT INTO transaction values (23232, 20221201, 400, 2020, 2642);
Query OK, 1 row affected (0.028 sec)
```

```
MariaDB [bank]> SELECT * FROM account where account_id=2020;
```

account_id	balance	type	IFSC	customer_id
2020	20400	Savings	2000	9382

```
1 row in set (0.001 sec)
```

## FRONT-END

The USER-INTERFACE is a python-library (streamlit) based application, that mainly consists of 3 roles ie. Customer, Employee and Admin.

Customers, Employees and the Admin can login into the accounts with their unique IDs.

The demonstration with screenshots is followed in the next slides.

P.T.O

Login Page

localhost:8501

## Login

ID

Role

☒ None

☐ Customer

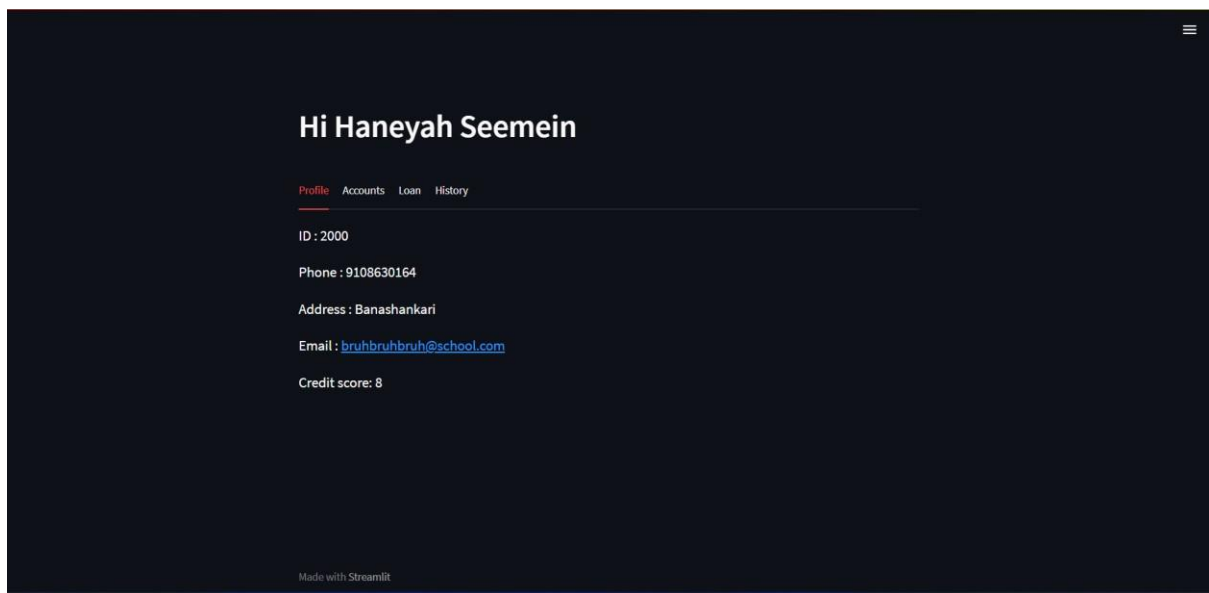
☐ Employee

☐ Admin

Made with Streamlit

Customer's interface

## Profile



The screenshot shows a web application interface with a dark blue background. At the top right, there is a hamburger menu icon. The main heading is "Hi Haneyah Seemein" in white. Below it, there is a navigation bar with four items: "Profile" (highlighted in red), "Accounts", "Loan", and "History". The profile information is listed below the navigation bar: "ID : 2000", "Phone : 9108630164", "Address : Banashankari", "Email : [bruhbruhbruh@school.com](mailto:bruhbruhbruh@school.com)", and "Credit score: 8". At the bottom left, it says "Made with Streamlit".

Hi Haneyah Seemein

Profile Accounts Loan History

ID : 2000

Phone : 9108630164

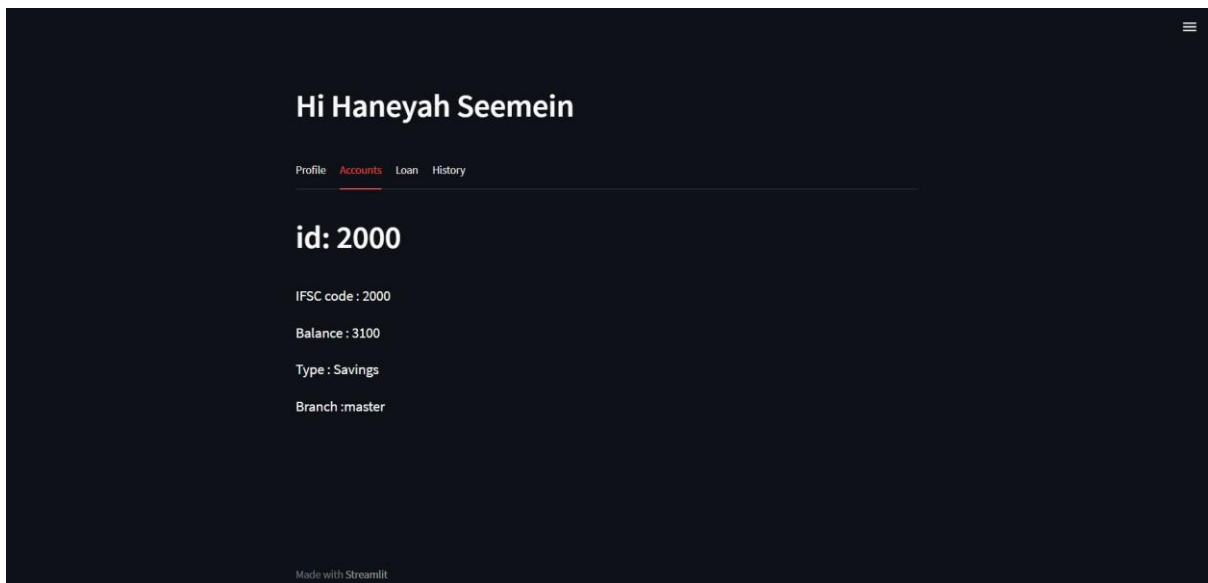
Address : Banashankari

Email : [bruhbruhbruh@school.com](mailto:bruhbruhbruh@school.com)

Credit score: 8

Made with Streamlit

## Account



The screenshot shows a web application interface with a dark blue background. At the top right, there is a hamburger menu icon. The main heading is "Hi Haneyah Seemein" in white. Below it, there is a navigation bar with four items: "Profile", "Accounts" (highlighted in red), "Loan", and "History". The account information is listed below the navigation bar: "id: 2000", "IFSC code : 2000", "Balance : 3100", "Type : Savings", and "Branch :master". At the bottom left, it says "Made with Streamlit".

Hi Haneyah Seemein

Profile Accounts Loan History

id: 2000

IFSC code : 2000


Balance : 3100

Type : Savings

Branch :master

Made with Streamlit

## Loan



## Hi Haneyah Seemein

[Profile](#)
[Accounts](#)
[Loan](#)
[History](#)

---

Loan ID : 2147483647

Amount : 140000

Time : 3

Type : Fund

Total Amount : 140000


Assisted employee name : Hazel Jills

Helpline : 7526358691

Made with Streamlit

Apps using your microphone:  
Your assistant is ready to respond.

### History



## Hi Haneyah Seemein

[Profile](#)
[Accounts](#)
[Loan](#)
[History](#)

---

### Recieved

**23990**

Amount: ₹300---Date : 20221125


**44505**

Amount: ₹1400---Date : 20221128

**47431**

Amount: ₹1400---Date : 20221125

**81361**



Amount: ₹800---Date : 20221125

### Sent

**23990**

Amount: 300

Date : 20221125

**81361**

Amount: 500

Date : 20221125

**87747**

Amount: -200000

Date : 20221125

## Employee's interface

Transaction

# Employee

HELLO Hazel Jills

[Transaction](#) [Loan](#) [Account](#) [Customers](#)

☒ Credit

☐ Debit

Customer's account ID

Customer's ID

Amount

transact

## Loan

[Transaction](#) [Loan](#) [Account](#) [Customers](#)

Amount

time(in years)

Type

customer's ID

branch's IFSC code

Apply

## Account

Transaction Loan **Account** Customers

Account ID  
0 - +

Amount  
200000 - +

Type  
Savings ▾

branch's IFSC code  
1000 - +

customer's ID  
0 - +

Create

## Customers

Employee

HELLO Hazel Jills

Transaction Loan Account **Customers**

### loans

	0	1	2	3	4	5
0	22112512	Sophronia Postins	Home	1164260229	380000	11
1	211251209	Hilary Welband	Home	3977766598	380000	11
2	211281154	Shawna Corbin	Home	1199864941	200000	11
3	2147483647	Haneyah Seemeln	Fund	9108630164	140000	11

Made with Streamlit

## Admin's Interface

Hire



HireFireDelete customerDelete accountCustom

Name

ID

0- +

Role

Branch Head

Points

0.00- +

Salary

10000- +

Phone

0- +

Email

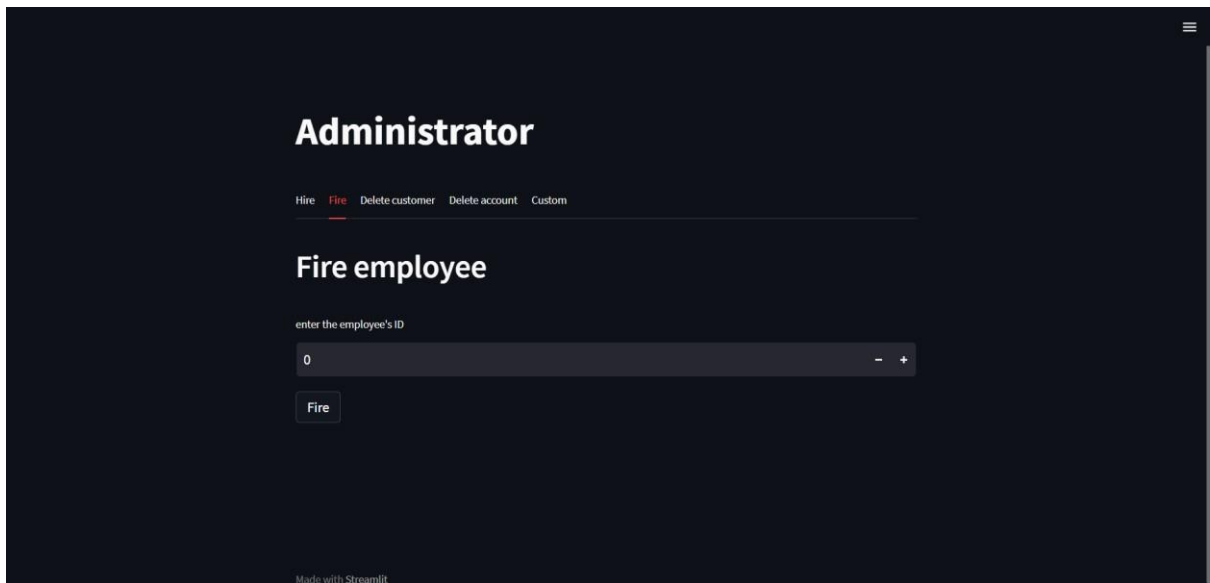
Address

IFSC code

1000- +

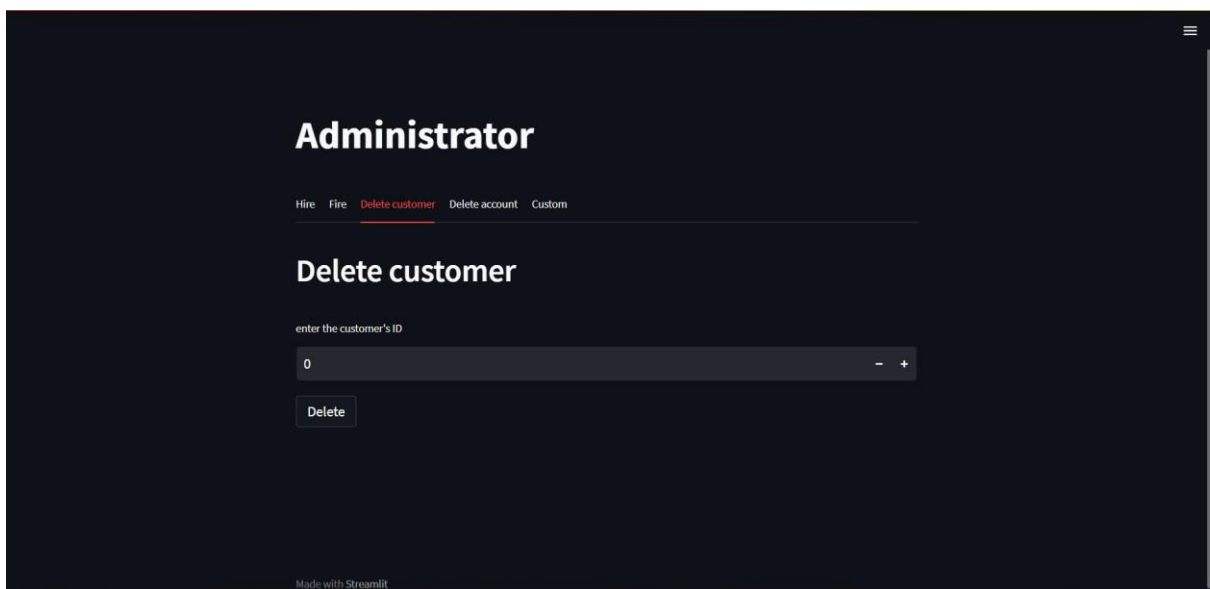
Submit

Fire



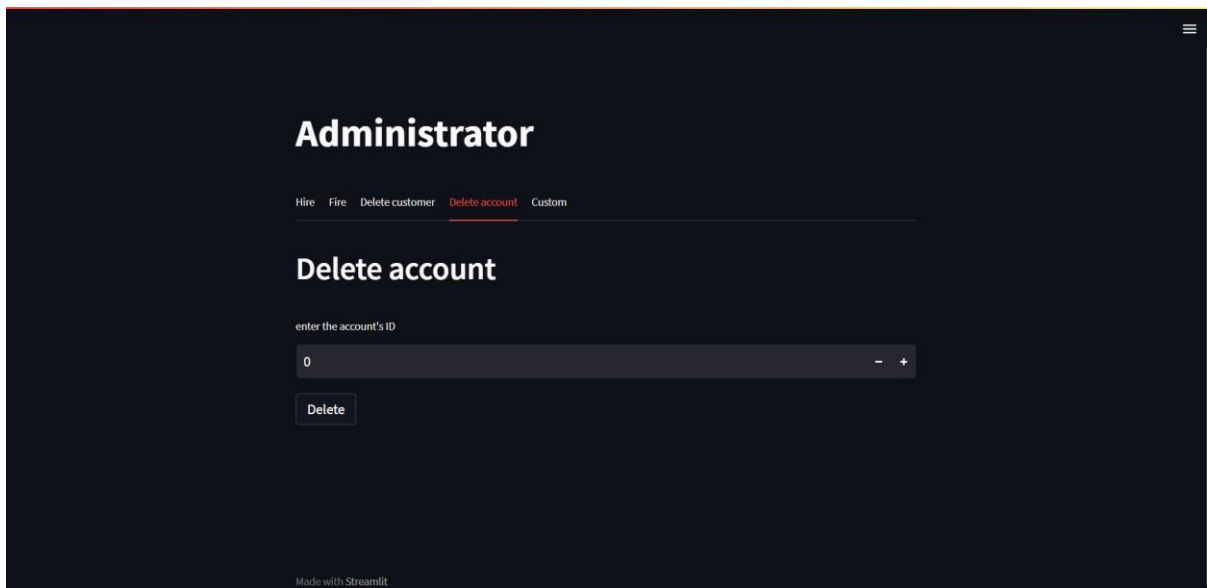
The screenshot shows a web application titled "Administrator" with a dark theme. A navigation bar at the top contains links: "Hire", "Fire" (highlighted in red), "Delete customer", "Delete account", and "Custom". Below the navigation bar, the main heading is "Fire employee". Underneath, there is a text input field labeled "enter the employee's ID" containing the number "0". To the right of the input field are minus and plus icons. Below the input field is a button labeled "Fire". At the bottom left, it says "Made with Streamlit".

Delete customer

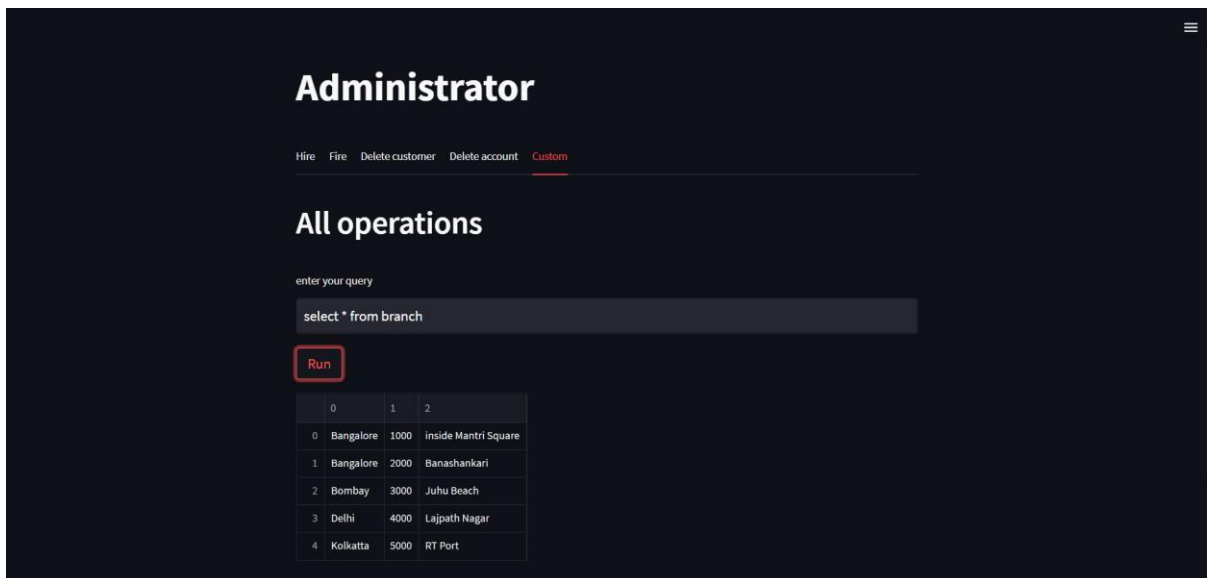


The screenshot shows the same "Administrator" web application, but the "Delete customer" link in the navigation bar is highlighted in red. The main heading is "Delete customer". The text input field is labeled "enter the customer's ID" and contains the number "0". To the right of the input field are minus and plus icons. Below the input field is a button labeled "Delete". At the bottom left, it says "Made with Streamlit".

Delete account



Custom



---

# THANK YOU

---