https://github.com/sanjith-s/airlinemanagement

Carvel

An one stop platform for your travel and logistics needs.

PROBLEM STATEMENT:

Traditionally, if a person wants to avail a service of an airline like travel, cargo etc., he/she has to undergo a tedious process. This involves manual processes of going to the airport directly or approaching middlemen such as the travel agent to avail the services which results in additional charges. The user cannot convey or receive the latest updates regarding the service he/she is availing. There is also a risk of mismanagement of data when the booking process is done through a physical mode. There is less security and it is not very user-friendly as well. Owing to these reasons, the process of physically booking to avail a service turned out to be exhausting.

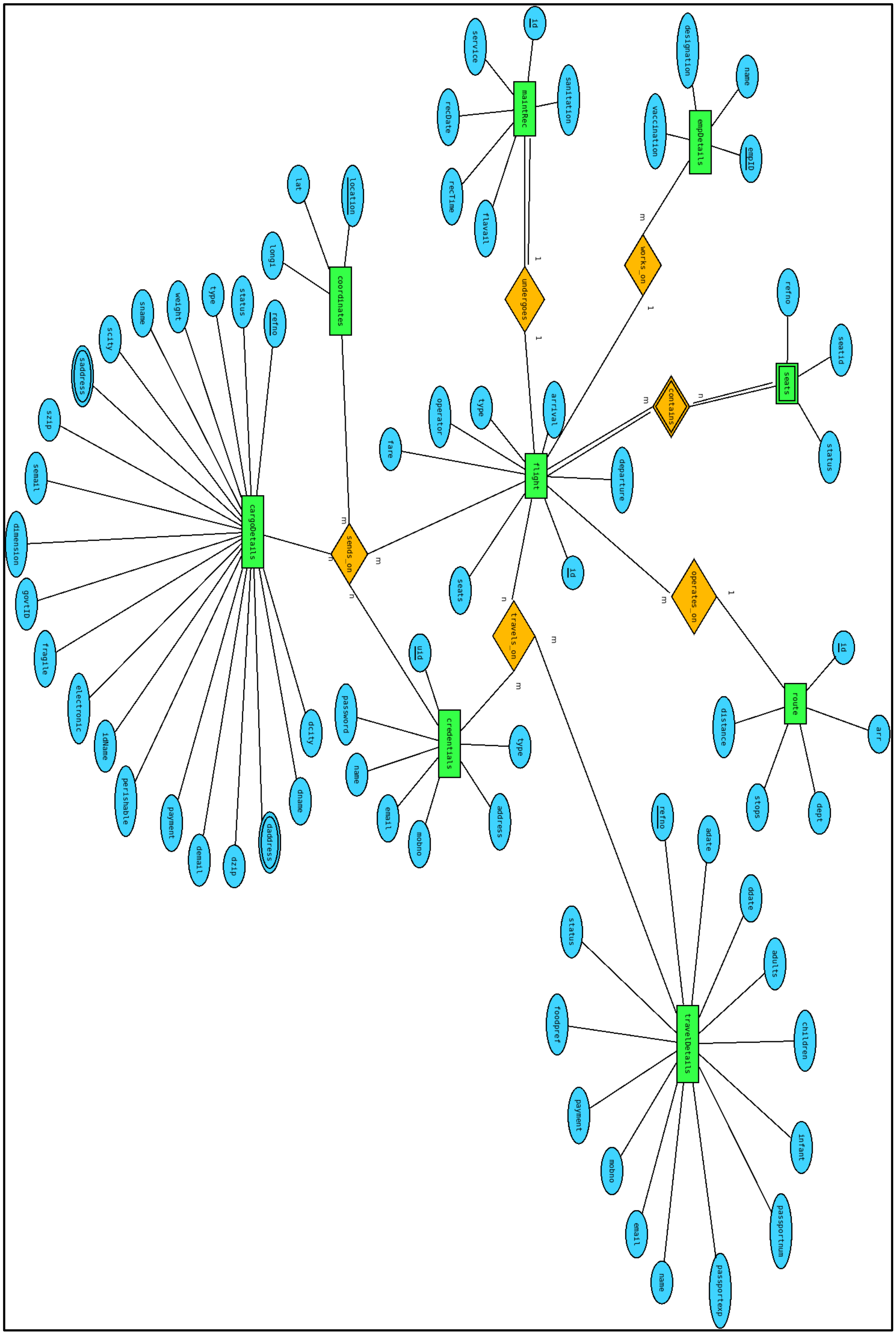
DESCRIPTION OF THE SOLUTION:

The proposed Airline Management Database System aims to computerize and automate the flight operations, ticket booking and cargo services provided by an airline. The ticket booking service will offer the cheapest and fastest routes based on the user preference. The user can manage his bookings where he/she can modify and view the booking history. The cargo service will enable the user to book his consignment and receive updates on the delivery of the same. The system will also provide catering services for the passengers.

IMPLEMENTATION:

The system is planned to be implemented as a Web Application focusing on a proper tradeoff between UX and functionality. The front-end will be implemented using ReactJS, CSS Bootstrap and Material UI frameworks. The backend will be implemented using NodeJS and MySQL for database management.

ER DIAGRAM:


id
service
sanitation
maintRec
recDate
recTime
flavail
undergoes
designation
name
empDetails
vaccination
empID
works_on
lat
location
longi
coordinates
refno
seatid
seats
contains
status
arrival
type
operator
fare
flight
departure
seats
id
sends_on
cargoDetails
sname
weight
type
status
refno
scity
saddress
szip
semail
dimension
govtID
fragile
electronic
idName
perishable
payment
demail
dzip
daddress
dname
dcity
travels_on
operates_on
uid
password
credentials
type
name
email
mobno
address
route
id
distance
arr
stops
dept
travelDetails
refno
adate
ddate
adults
children
infant
passportnum
passportexp
name
email
mobno
payment
foodpref
status

TABLES:

cargoDest (refno, dname, addrl1, addrl2, addrl3, dest, dzip, email)

cargoDetails (refno, weight, dimension, fragile, electronic, perishable, userid, type, status, govtID, idName)

cargoSrc (refno, sname, addrl1, addrl2, addrl3, src, szip, email)

coordinates (id, location, lat, longi)

credentials (uid, email, password, name, mobno, address, type)

empDetails (empID, name, designation, vaccination)

flight (id, operator, departure, arrival, seats, duration, type, routeid, fare)

maintRec (id, flightid, sanitation, service, recdate, rectime, flavail)

route (id, dept, arr, stops, distance)

seats (flightid, seatid, status, refno)

travelDetails (refno, uid, adate, ddate, flightno, adults, children, infant, passportnum, passportexp, name, email, mobno, payment, foodpref

QUERIES:

- INSERT INTO empDetails (empID, name, designation, vaccination) VALUES (?,?,?,?);

- INSERT INTO maintRec (flightid, sanitation, service, recdate, rectime, flavail) VALUES (?,?,?,?,?,?);

- SELECT * FROM credentials WHERE email=?;

- SELECT * FROM cargoDetails ORDER BY refno DESC LIMIT 1;

- INSERT INTO route (id, dept, arr, stops, distance) VALUES (?,?,?,?,?);

- SELECT lat, longi FROM coordinates WHERE location=?;

- INSERT INTO cargoSrc (name, addrl1, addrl2, addrl3, src, zip, email) VALUES (?,?,?,?,?,?,?);

- INSERT INTO cargoDest (name, addrl1, addrl2, addrl3, dest, zip, email) VALUES (?,?,?,?,?,?,?);

- INSERT INTO cargoDetails (type, weight, dimension, fragile, electronic, perishable, userid, govtID, idName, status) VALUES (?,?,?,?,?,?,?,?,?,?);

- INSERT INTO flight VALUES (?,?,?,?,?,?,?,?,?);

- INSERT INTO seats (flightid, seatid, status) VALUES (?,?,?);

- SELECT * FROM route WHERE dept=? AND arr=?;

- SELECT * FROM flight WHERE routeid=? AND type=? ORDER BY fare
- ASC, duration ASC ;

- SELECT * FROM seats WHERE flightid=?;

- SELECT * FROM flight WHERE id=?;
- INSERT INTO flight VALUES (?,?,?,?,?,?,?,?,?);

- SELECT * FROM empDetails WHERE empID=?;

- UPDATE empDetails SET name=?, designation=?, vaccination=? WHERE empID=?;

- SELECT * FROM flight WHERE id=?;

- UPDATE flight SET operator=?, departure=?, arrival=?, duration=?, seats=?, type=?, fare=?, routeid=? WHERE id=?;

- DELETE FROM seats WHERE flightid=?;

- INSERT INTO seats (flightid, seatid, status) VALUES (?,?,?);

- SELECT * FROM route WHERE id=?;

- UPDATE route SET dept=?, arr=?, stops=?, distance=? WHERE id=?;

- SELECT * FROM maintRec WHERE id=?;

- UPDATE maintRec SET sanitation=?, service=?, recdate=?, rectime=?, flavail=? WHERE id=?;

- DELETE FROM empDetails WHERE empID=?;

- DELETE FROM flight WHERE id=?;

- DELETE FROM maintRec WHERE id=?;

- DELETE FROM route WHERE id=?;

- UPDATE cargoDetails SET status=? WHERE refno=?;

- SELECT refno,status FROM cargoDetails WHERE refno=?;

- SELECT sname, src, szip, dname, dest, dzip, weight, type, dimension, status FROM cargoDetails, cargoSrc, cargoDest WHERE userid=? AND cargoSrc.refno=cargoDest.refno AND cargoSrc.refno=cargoDetails.refno;

- SELECT refno, adate, ddate, payment, flightno, dept, arr, operator FROM travelDetails T, flight F, route R WHERE flightno=F.id AND routeid=R.id AND uid=?;

TRIGGERS:

- **TRIGGER 1:**

```
DROP TRIGGER IF EXISTS `airlineSystem`.`maintRec_AFTER_INSERT`;
DELIMITER $$
USE `airlineSystem`$$
CREATE DEFINER=`root`@`localhost` TRIGGER `maintRec_AFTER_INSERT`
AFTER INSERT ON `maintRec` FOR EACH ROW BEGIN
if NEW.flavail='NO' then
                UPDATE flight SET status="NO" WHERE id=NEW.flightid;
        else
                UPDATE flight SET status="YES" WHERE id=NEW.flightid;
   end if;
END$$
DELIMITER ;
```

- **TRIGGER 2:**

```
CREATE DEFINER = CURRENT_USER TRIGGER `airlineSystem`.`createseats`
AFTER INSERT ON `flight` FOR EACH ROW
BEGIN
        DECLARE i INT unsigned DEFAULT 1;
   SET i=1;
   createseat: LOOP
                INSERT INTO seats values(NEW.id, i, 'YES', -1);
      SET i=i+1;
   IF i <= NEW.seats THEN
     ITERATE createseat;
   END IF;
   LEAVE createseat;
  END LOOP createseat;
END
```

- **TRIGGER 3:**

```
CREATE DEFINER=`root`@`localhost` TRIGGER `flight_AFTER_INSERT`
AFTER INSERT ON `flight` FOR EACH ROW BEGIN
UPDATE flight SET status='YES' WHERE id=NEW.id;
END
```

- **TRIGGER 4:**

```
CREATE DEFINER=`root`@`localhost` TRIGGER `deleteseat` AFTER DELETE
ON `flight` FOR EACH ROW BEGIN
        DELETE FROM seats WHERE flightid=OLD.id;

END
```

- **TRIGGER 5:**

```
CREATE DEFINER=`root`@`localhost` TRIGGER `deleteflight` AFTER DELETE
ON `route` FOR EACH ROW BEGIN
        DELETE FROM flight WHERE routeid=OLD.id;
END
```

PROCEDURES:

```
USE `airlineSystem`;
DROP procedure IF EXISTS `emailFunc`;

DELIMITER $$
USE `airlineSystem`$$
CREATE PROCEDURE `emailFunc`(in ID int)
BEGIN
        SELECT * FROM credentials  WHERE uid=id;
END$$
DELIMITER ;
```

CONCLUSION:

We have an implemented an Airline Management System which supports both cargo and travel operations. Through this project, we have learned how to integrate a database such as SQL with the front-end and back-end of the website. We have also learnt how to approach real-life problems and solve them.

FUTURE SCOPE:

The proposed system aims to provide a unified web application which encompasses the services offered by an airline in a user-friendly interface. The system performs the following functions:

1. Ticket booking portal
2. Maintenance of travel history
3. Cargo booking portal
4. Flight and cargo tracking
5. Management of flight operations
6. Provision of discounts based on travel miles
7. Scalability