```
var express = require('express');
var passport = require('passport');
var Strategy = require('passport-facebook').Strategy;



var mysql      = require('mysql');
var connection = mysql.createConnection({
  host     : 'localhost',
  user     : 'root',
  password : '',
  database : 'simofi'
});

connection.connect();






// Configure the Facebook strategy for use by Passport.
//
// OAuth 2.0-based strategies require a `verify` function which receives the
// credential (`accessToken`) for accessing the Facebook API on the user's
// behalf, along with the user's profile.  The function must invoke `cb`
// with a user object, which will be set at `req.user` in route handlers after
// authentication.
passport.use(new Strategy({
    clientID:'260049161073172',
    clientSecret:'fc848418eb7426bde5fffffb50555a90',
    callbackURL: 'http://localhost:3000/login/facebook/return',





  },
  function(req,accessToken, refreshToken, profile, cb) {
    // In this example, the user's Facebook profile is supplied as the user
    // record.  In a production-quality application, the Facebook profile should
    // be associated with a user record in the application's database, which
    // allows for account linking and authentication with other identity
    // providers.

    // clientID, clientSecret and callbackURL
   // profileFields: ['email']

 var user = {


        'name' : profile.displayName,
        'id'   : profile.id,
        'email':profile.emails[0].value



        }



connection.query('insert into fb_posts (Id,UserName) values('+user.id+','"'+user.name+'"')', function(err,
rows, fields) {
  if (!err)
    console.log('The solution is: ', rows);
  else
    console.log('Error while performing Query.');

});



console.log(user.displayName);
console.log(user.id);
console.log(user.email)
//console.log(JSON.stringify(user.name));



    return cb(null, profile);
  }));


passport.serializeUser(function(user, cb) {
  cb(null, user);
});

passport.deserializeUser(function(obj, cb) {
  cb(null, obj);
});


// Create a new Express application.
var app = express();
```

```
// Configure view engine to render EJS templates.

app.set('views', __dirname + '/views');

app.set('view engine', 'ejs');


// Use application-level middleware for common functionality, including

// logging, parsing, and session handling.

app.use(require('morgan')('combined'));

app.use(require('cookie-parser')());

app.use(require('body-parser').urlencoded({ extended: true }));//All middlewares will populate the
req.body property with the parsed body, or an empty object ({}) if there was no body to parse (or an
error was returned

app.use(require('express-session')({ secret: 'keyboard cat', resave: true, saveUninitialized: true }));


// Initialize Passport and restore authentication state, if any, from the

// session.

app.use(passport.initialize());

app.use(passport.session());



// Define routes.

app.get('/',

  function(req, res) {

    res.render('home', { user: req.user });

  });


app.get('/login',

  function(req, res){

    res.render('login');
  });

app.get('/login/facebook',
  passport.authenticate('facebook'));

app.get('/login/facebook/return',
  passport.authenticate('facebook', { failureRedirect: '/login' }),
  function(req, res) {



    res.redirect('/');

  });

app.get('/profile',
  require('connect-ensure-login').ensureLoggedIn(),
  function(req, res){
    res.render('profile', { user: req.user });
  });

app.listen(3000);
```