CSE370 : Database Systems
Project Report
Project Title : Marriage Registration Portal

Group No : 11, CSE370 Lab Section : 04, Summer 2023		
ID	**Name**	**Contribution**
21301284	Khawaja Ehsun Ul Hawak	Everything

Table of Contents

Introduction

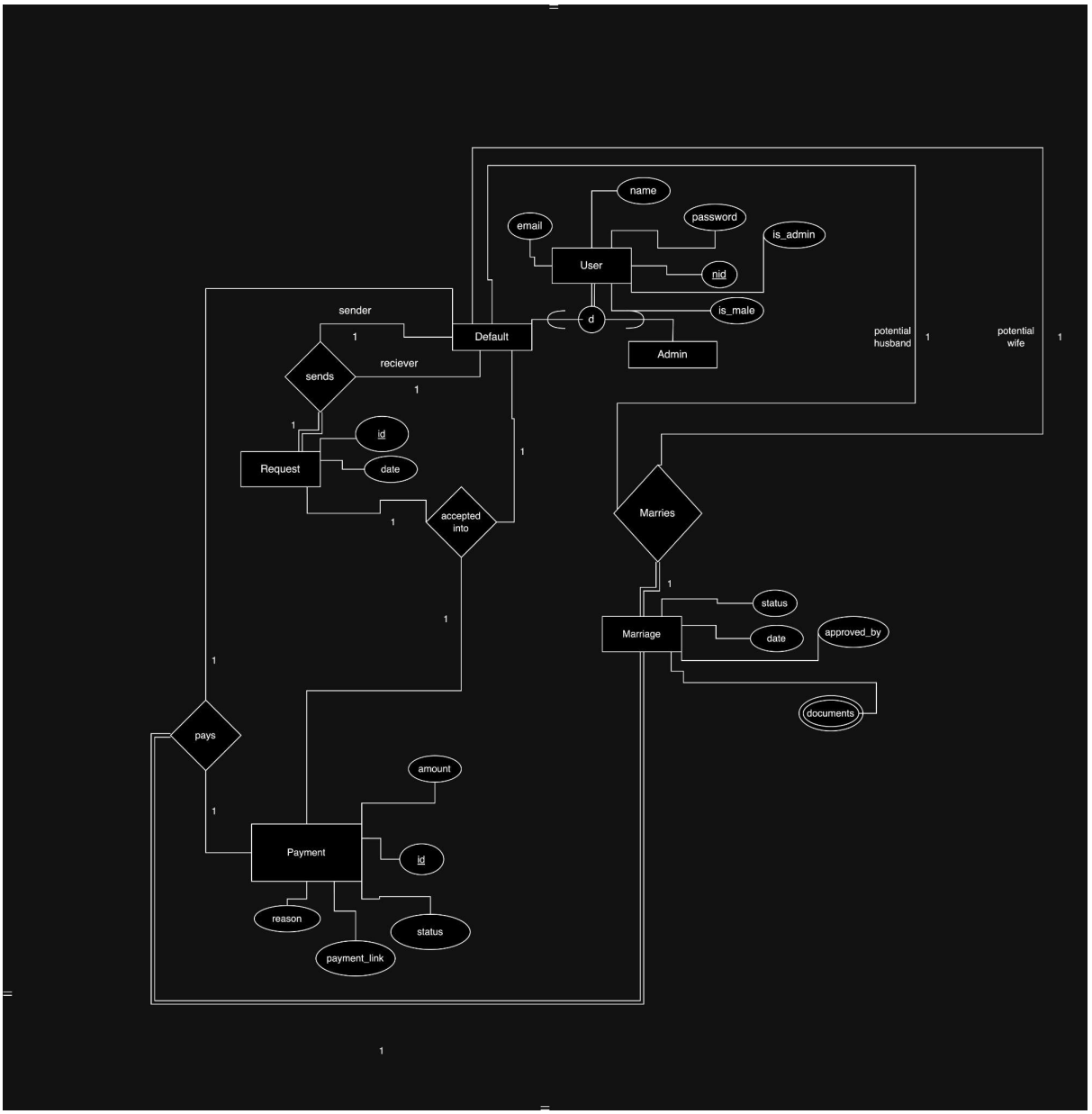
Introducing our advanced Marriage Registration Portal, designed to streamline the marriage process. In today's fast-paced world, couples deserve an efficient and easy way to make their unions official. This report delves into the reasons behind creating the portal, highlighting its crucial role in updating essential legal documentation. It finds a balance between tradition and innovation, providing a user-friendly digital platform for registering marriages. Beyond convenience, accurate registration has significant legal and societal implications, safeguarding spousal rights and ensuring the authenticity of family ties. The report underscores the portal's ability to lessen administrative burdens, minimize errors, and improve transparency—a step towards effective governance and the protection of individual rights.

Project Features

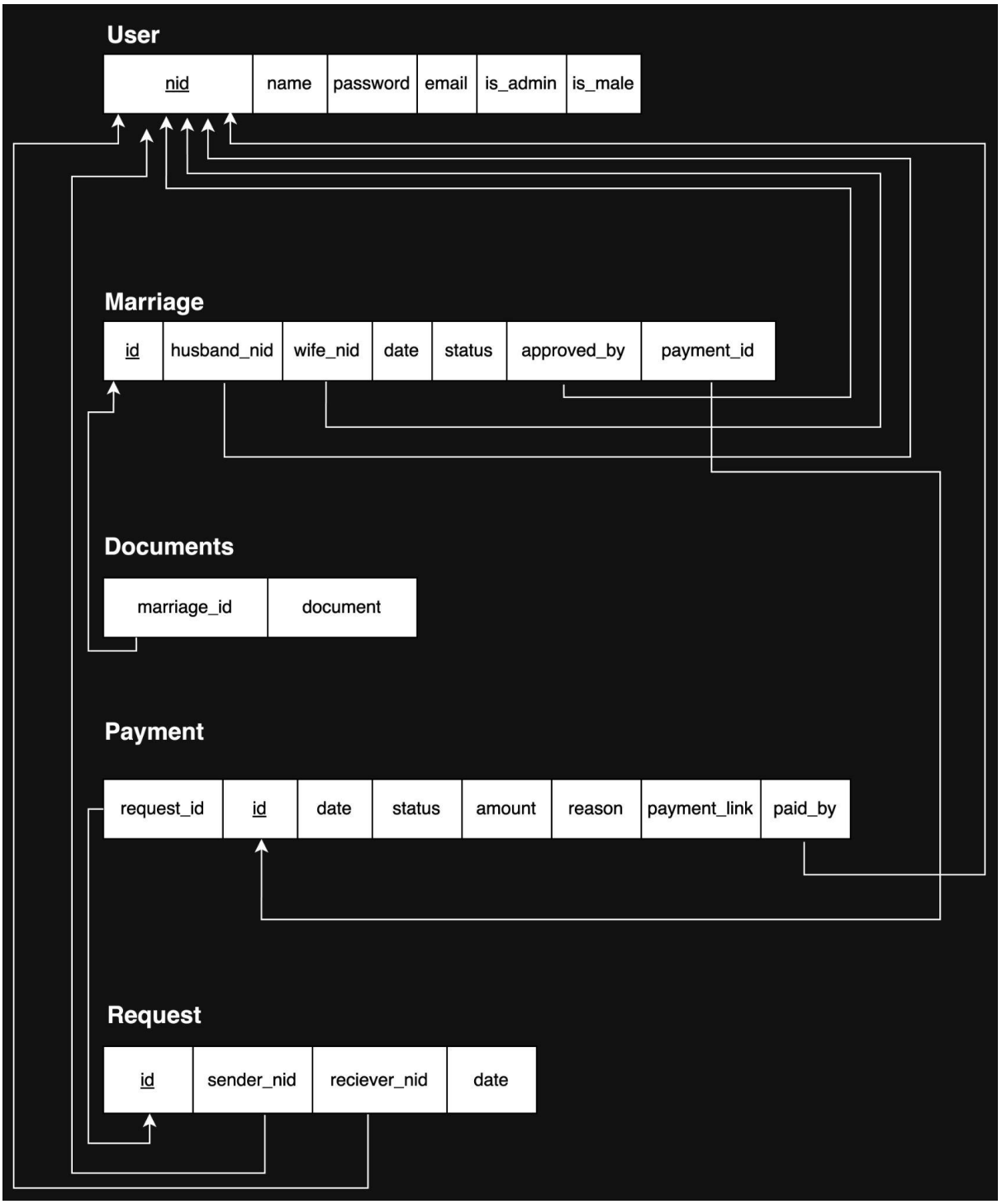
The web application contains features necessary to initiate and divorce marriages and everything in between.
The features are listed below:

1. **Authentication:** Each citizen of the country can register their account on the portal using their nid number to be available for marriage requests. They can later login with the password they set in the registration.
2. **Marriage Requests:** Marriages requests are like friend requests that can be send by both the male and female citizens to each other(if both of them do not have a partner). Once a request is made, the other party will have to accept the request and proceed with the necessary payment to advance the request into an incomplete marriage.
3. **Marriages:** A marriage starts as an incomplete marriage. ‘Incomplete’ implies that the marriage is still not complete and has not been submitted to the admins for approval. To complete it, one has to upload the necessary documents from the portal. After the documents have been submitted, the marriage status changes into unapproved.
4. **Admin:** Admins have two tasks, one is to approve or reject the marriages that are awaiting approval, other is to initiate the divorce when ordered from court. Users with admin access can easily see the documents and approve/reject the marriage. They can also search for marriages using the nid of husband and wife and change their marriages status to divorce.
5. **Payments:** As discussed in other features, payments are a crucial part and feature of the application.

ER/EER Diagram



Schema Diagram

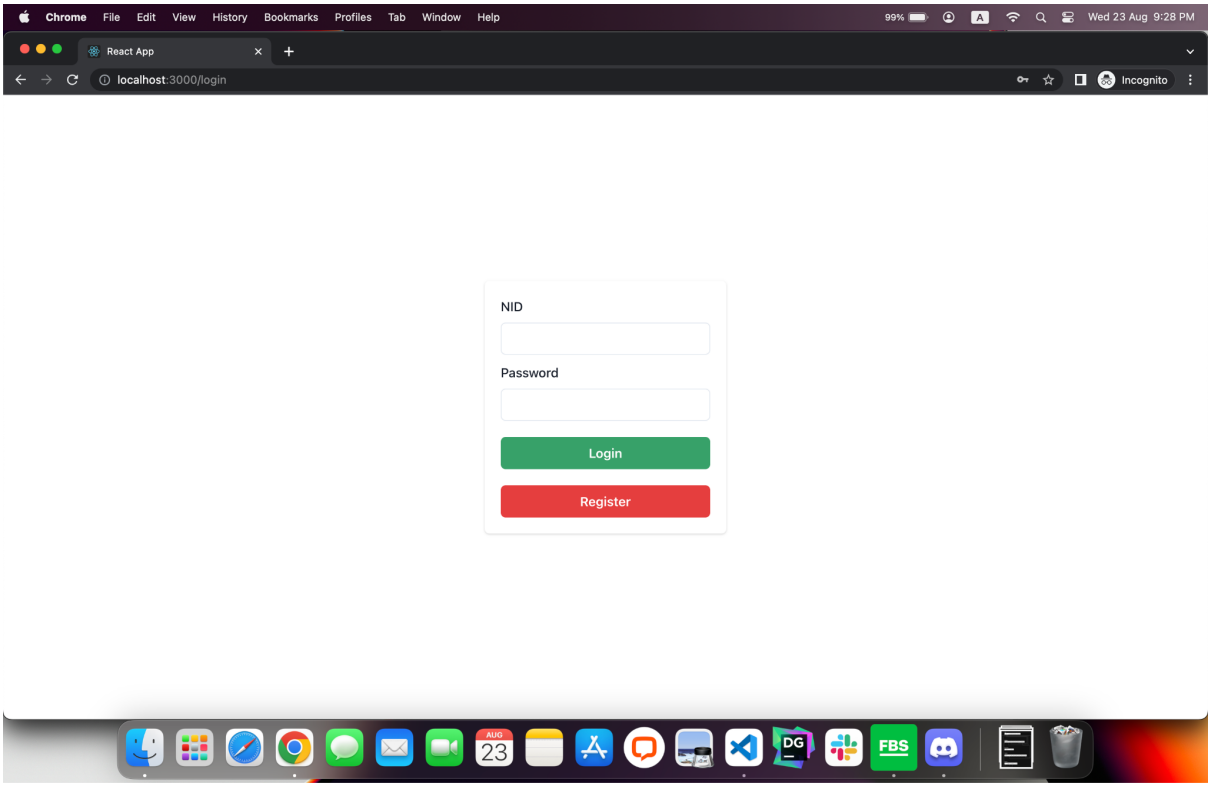


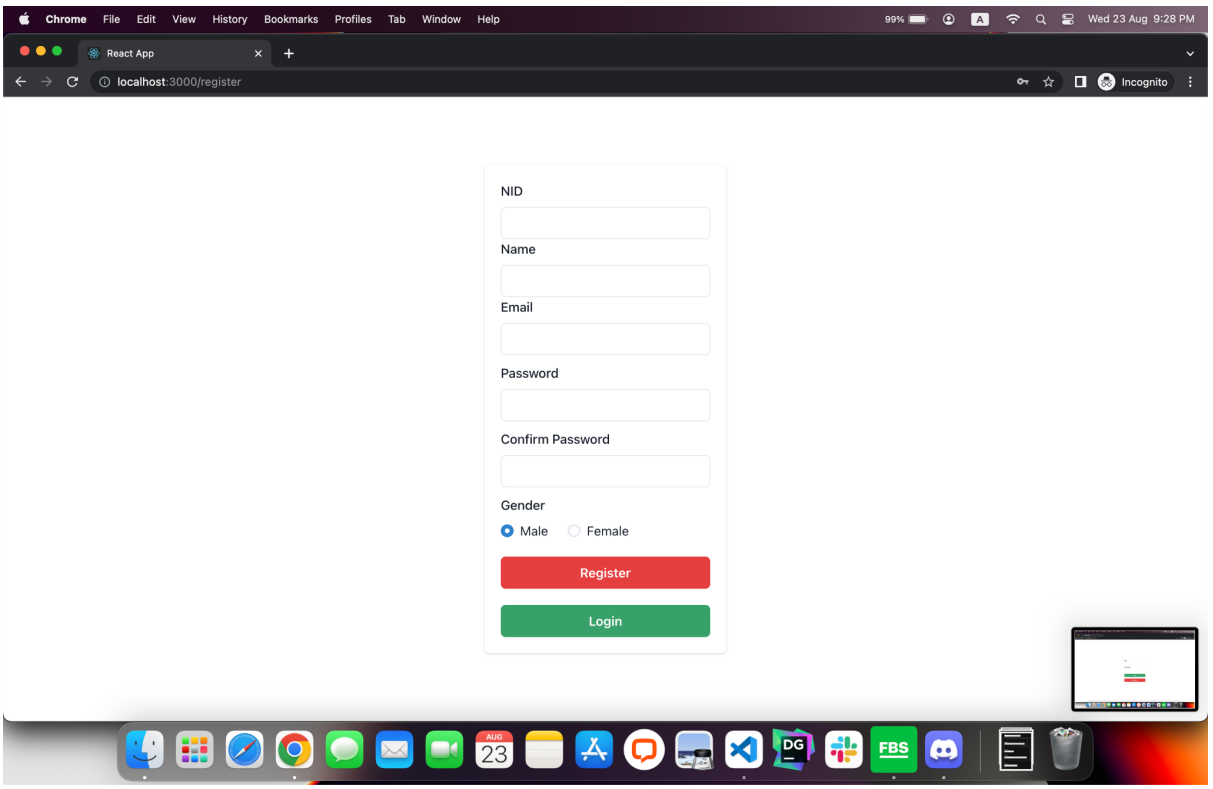
Frontend Development

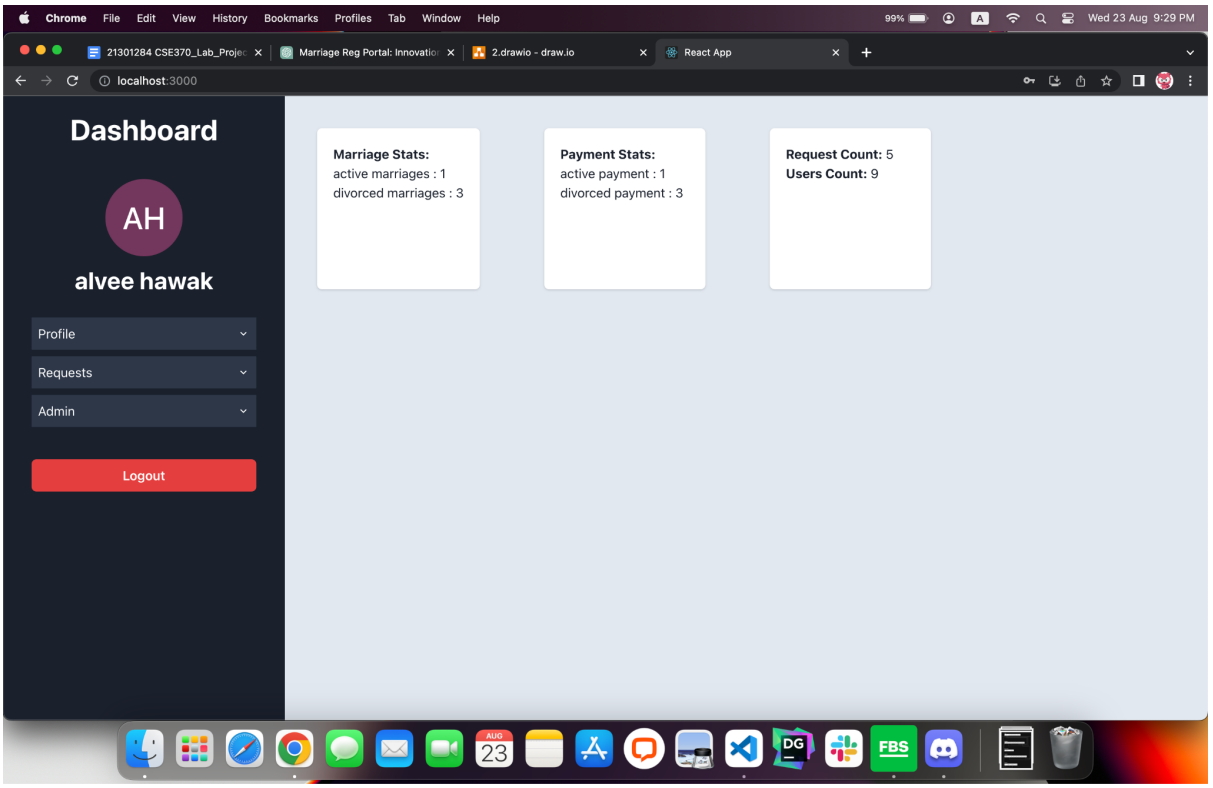
Frontend has been coded in javascript using react.js, chakra ui and other nodejs libraries.

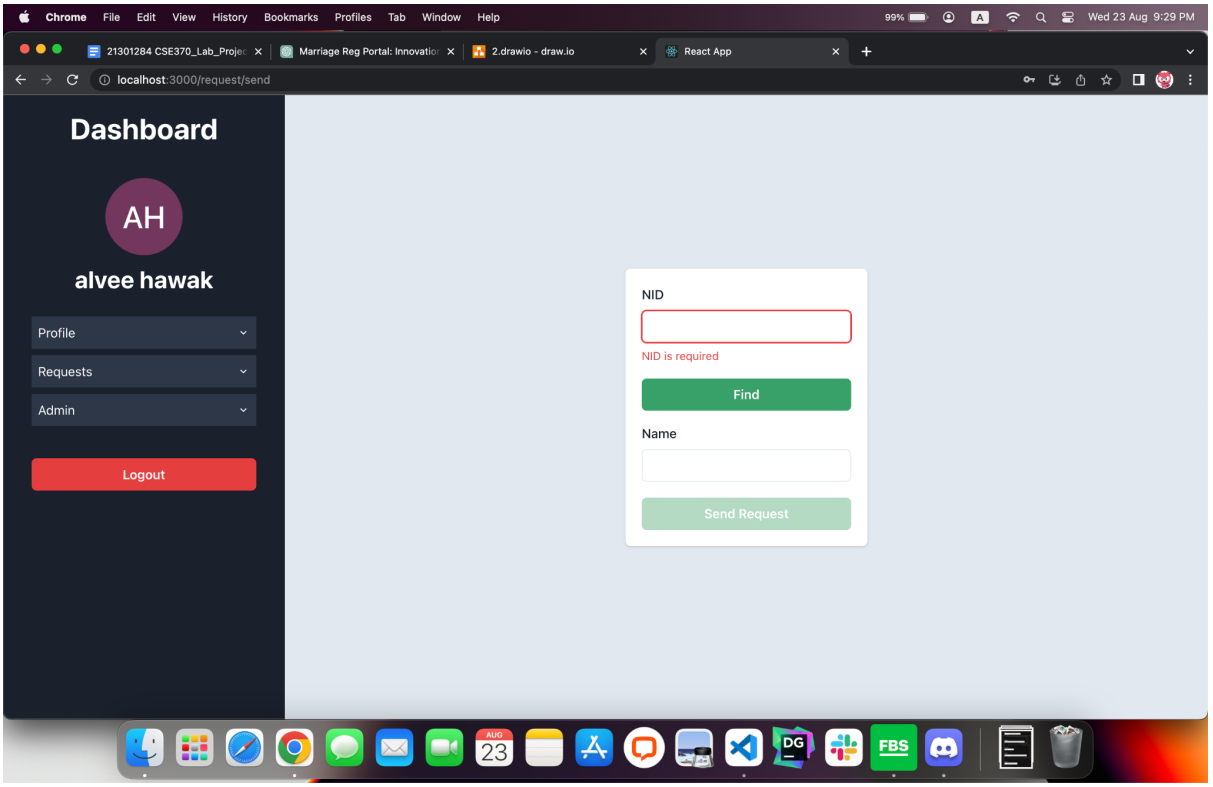
Separate pages has been coded for login, registration, sending requests, seeing all the recieved requests(payments are paid in this screen aswell), checking all the marriages of the users(uploading documents are done here aswell), 2 admin pages to do divorce and approving marriages.

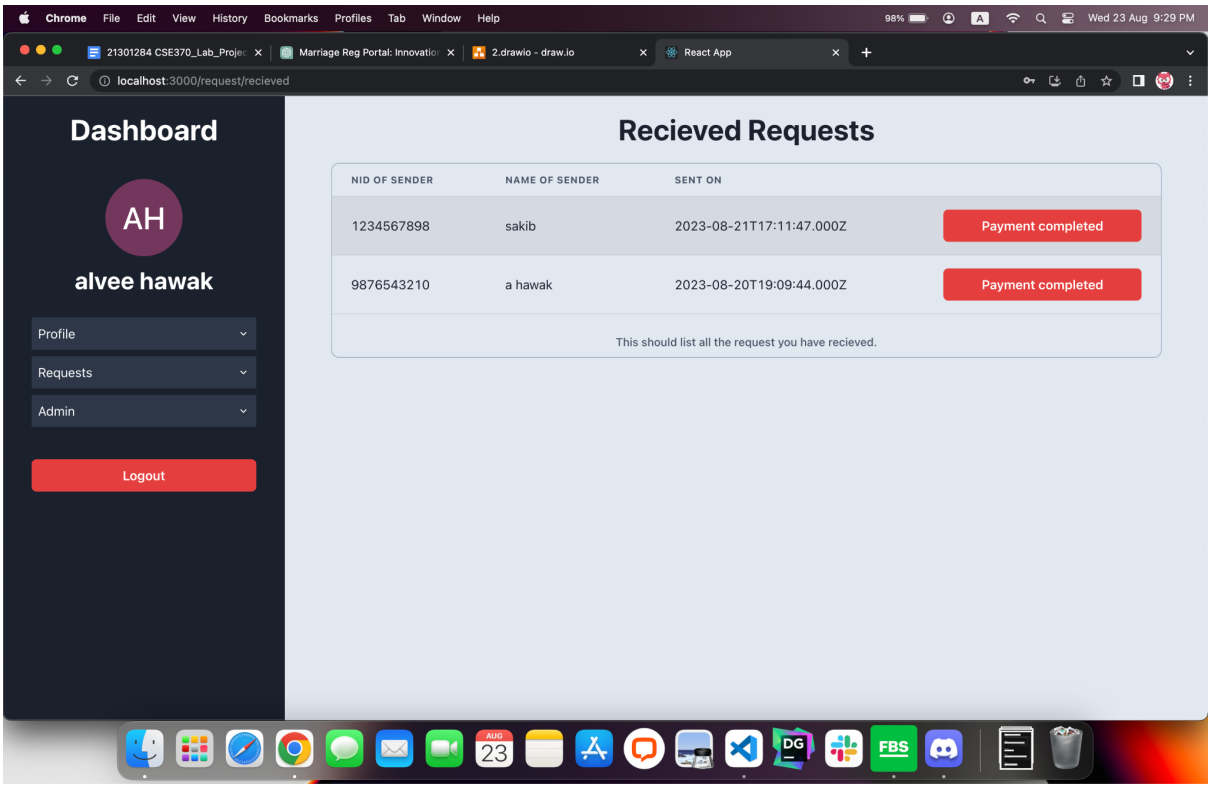
Here are some screenshots of the project's frontend:

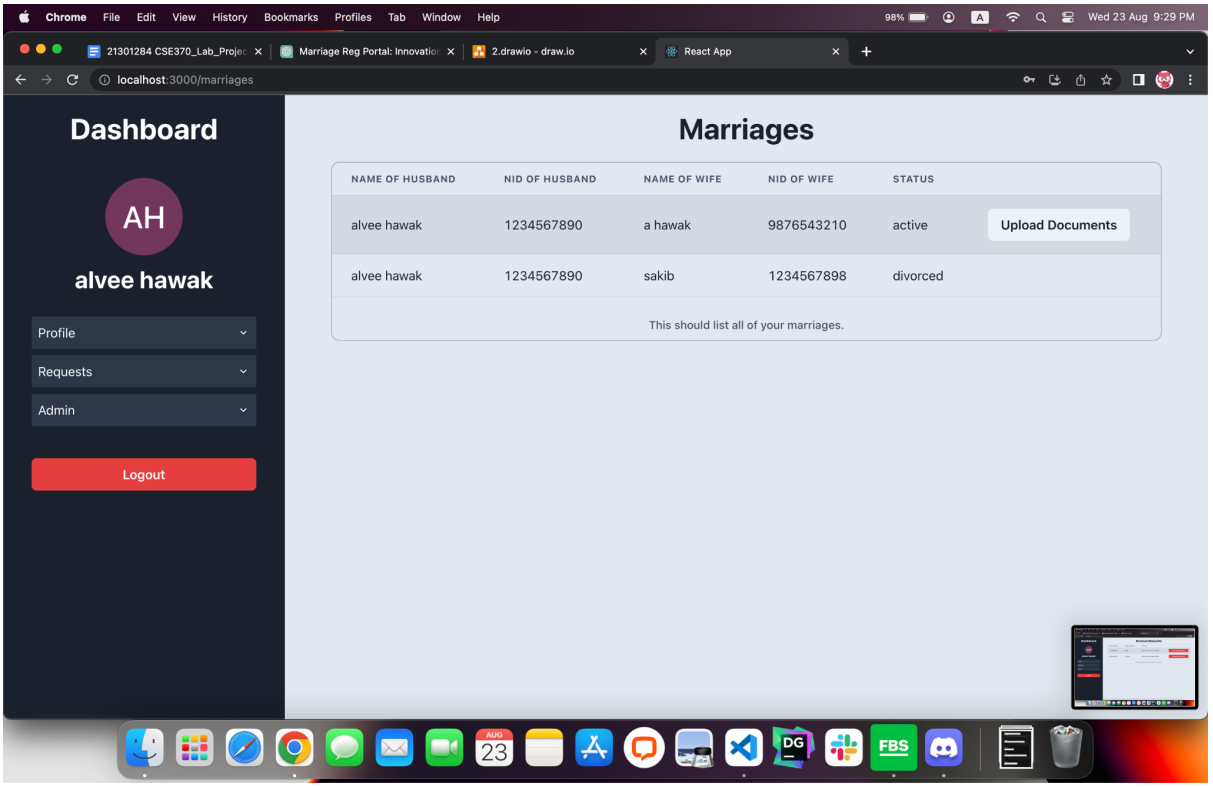


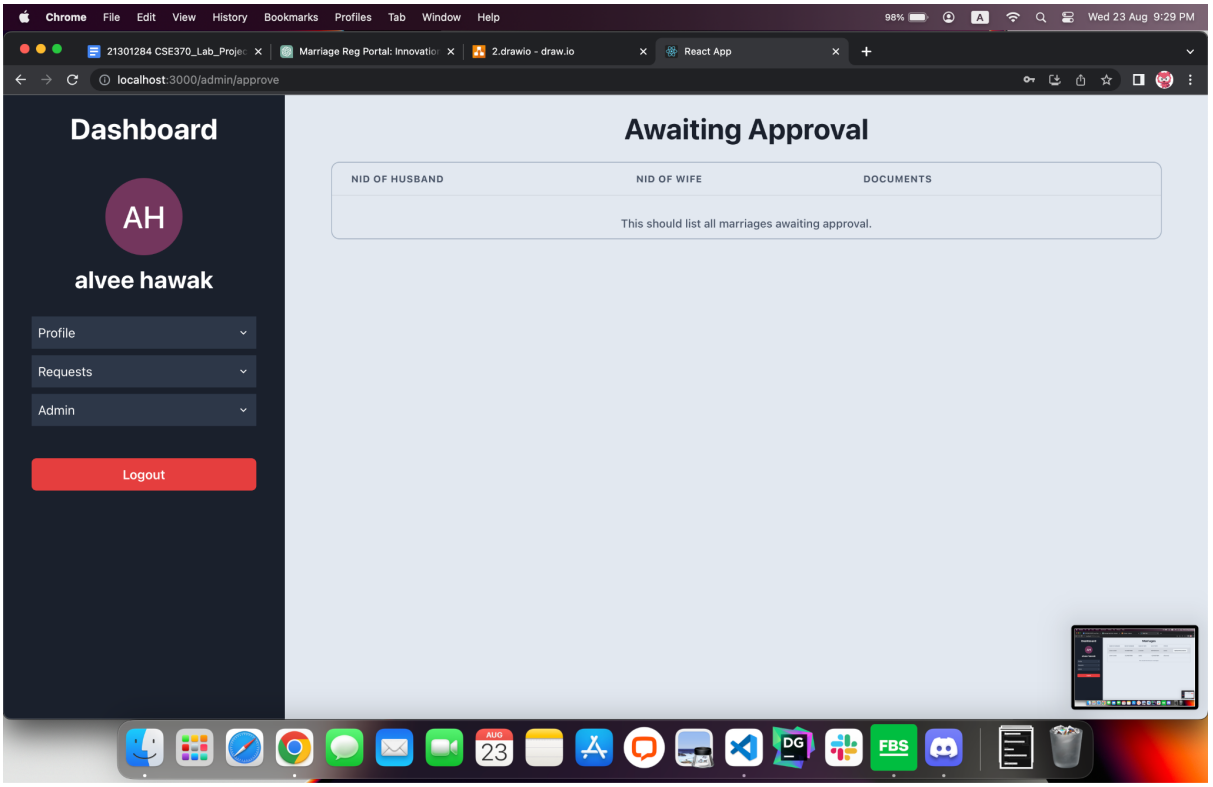

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 23 Aug 9:28 PM
React App
localhost:3000/register
Incognito
NID
Name
Email
Password
Confirm Password
Gender
Male
Female
Register
Login

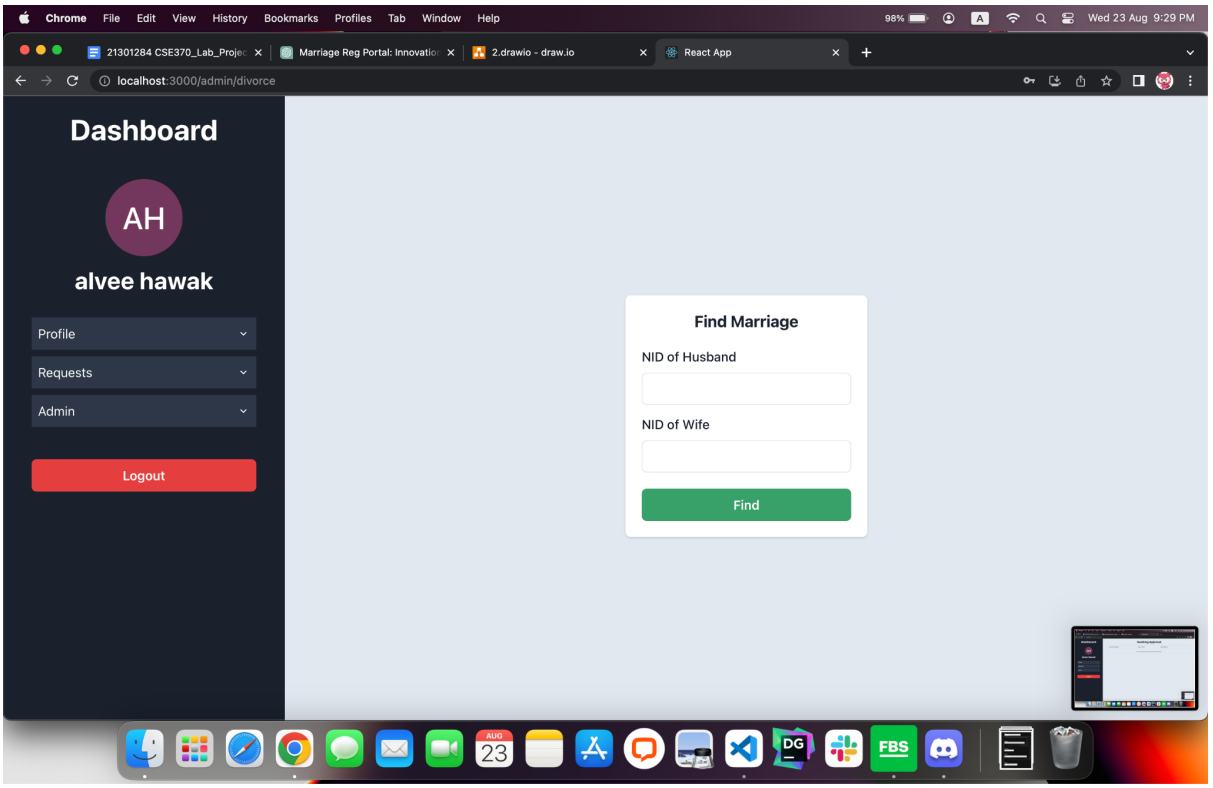

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 23 Aug 9:29 PM
21301284 CSE370_Lab_Proje
Marriage Reg Portal: Innovatio
2.drawio - draw.io
React App
localhost:3000
Dashboard
AH
alvee hawak
Profile
Requests
Admin
Logout
Marriage Stats:
active marriages : 1
divorced marriages : 3
Payment Stats:
active payment : 1
divorced payment : 3
Request Count: 5
Users Count: 9


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 23 Aug 9:29 PM
21301284 CSE370_Lab_Proje
Marriage Reg Portal: Innovatio
2.drawio - draw.io
React App
localhost:3000/request/send
Dashboard
AH
alvee hawak
Profile
Requests
Admin
Logout
NID
NID is required
Find
Name
Send Request


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 23 Aug 9:29 PM
21301284 CSE370_Lab_Proje
Marriage Reg Portal: Innovatio
2.drawio - draw.io
React App
localhost:3000/request/recieved
Dashboard
AH
alvee hawak
Profile
Requests
Admin
Logout
Recieved Requests
NID OF SENDER
NAME OF SENDER
SENT ON
1234567898
sakib
2023-08-21T17:11:47.000Z
Payment completed
9876543210
a hawak
2023-08-20T19:09:44.000Z
Payment completed
This should list all the request you have recieved.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 23 Aug 9:29 PM
localhost:3000/marriages
Dashboard
AH
alvee hawak
Profile
Requests
Admin
Logout
Marriages
NAME OF HUSBAND
NID OF HUSBAND
NAME OF WIFE
NID OF WIFE
STATUS
alvee hawak
1234567890
a hawak
9876543210
active
Upload Documents
alvee hawak
1234567890
sakib
1234567898
divorced
This should list all of your marriages.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 23 Aug 9:29 PM
21301284 CSE370_Lab_Proje
Marriage Reg Portal: Innovatio
2.drawio - draw.io
React App
localhost:3000/admin/approve
Dashboard
AH
alvee hawak
Profile
Requests
Admin
Logout
Awaiting Approval
NID OF HUSBAND
NID OF WIFE
DOCUMENTS
This should list all marriages awaiting approval.

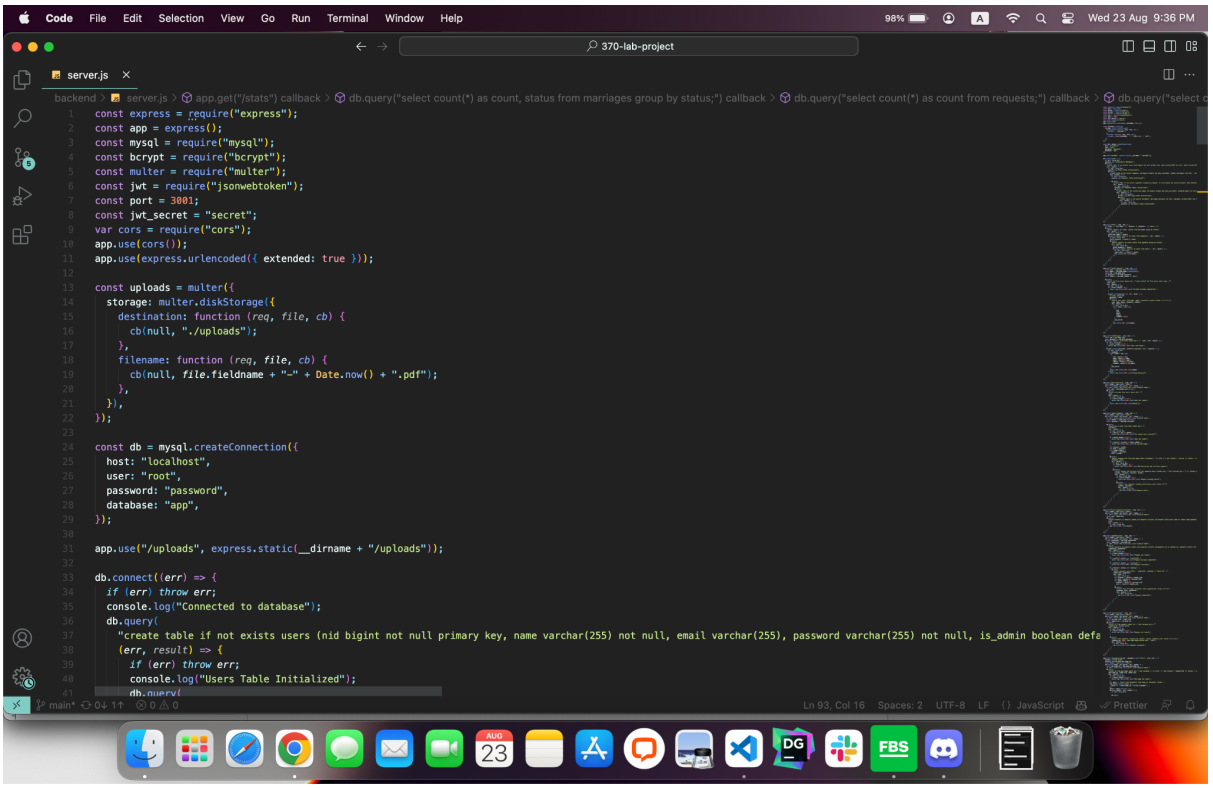

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 23 Aug 9:29 PM
21301284 CSE370_Lab_Proje
Marriage Reg Portal: Innovatio
2.drawio - draw.io
React App
localhost:3000/admin/divorce
Dashboard
AH
alvee hawak
Profile
Requests
Admin
Logout
Find Marriage
NID of Husband
NID of Wife
Find

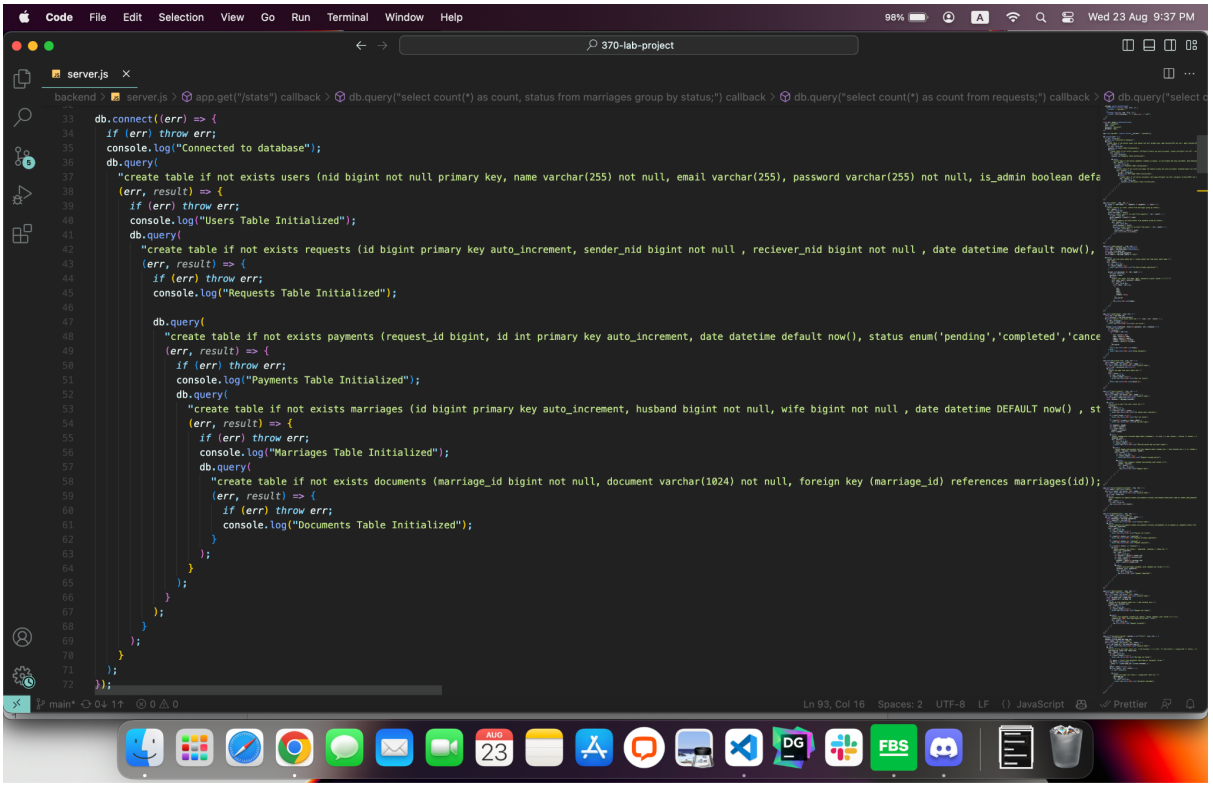
Backend Development

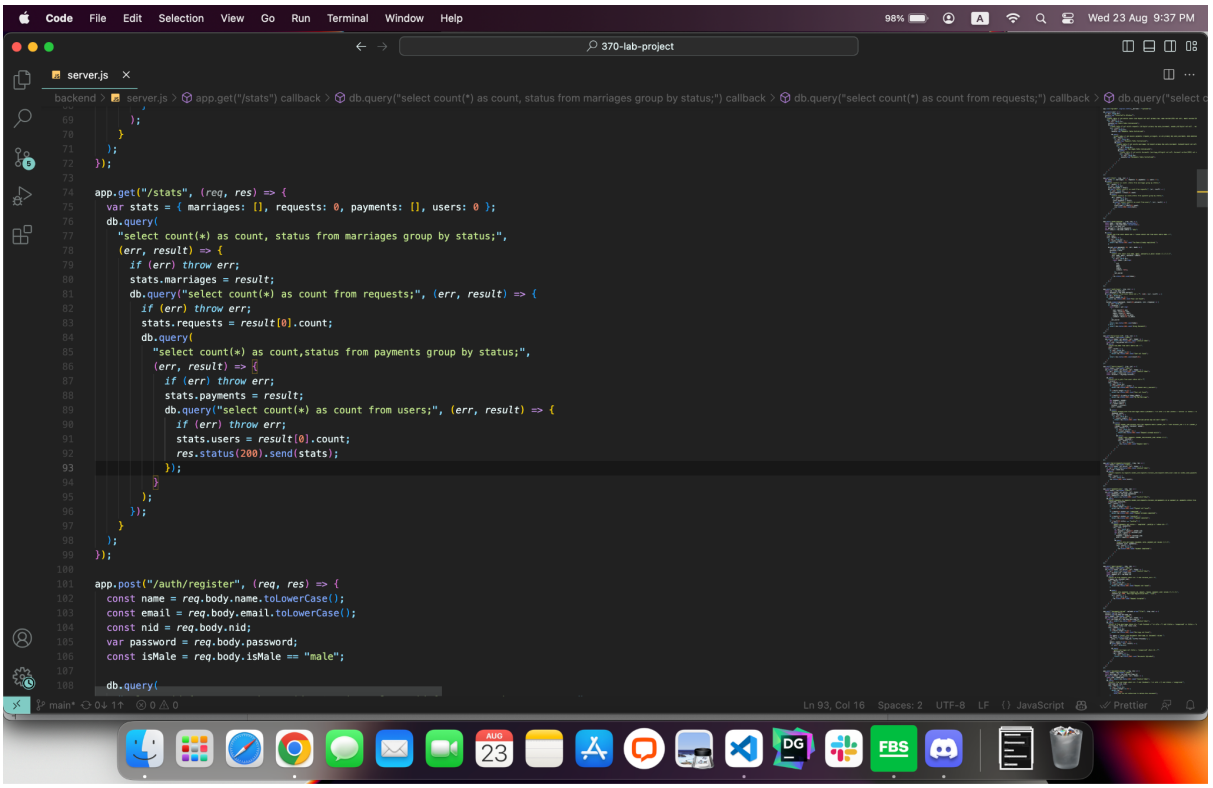
Backend has been coded with expressjs, mysql and other nodejs libraries.

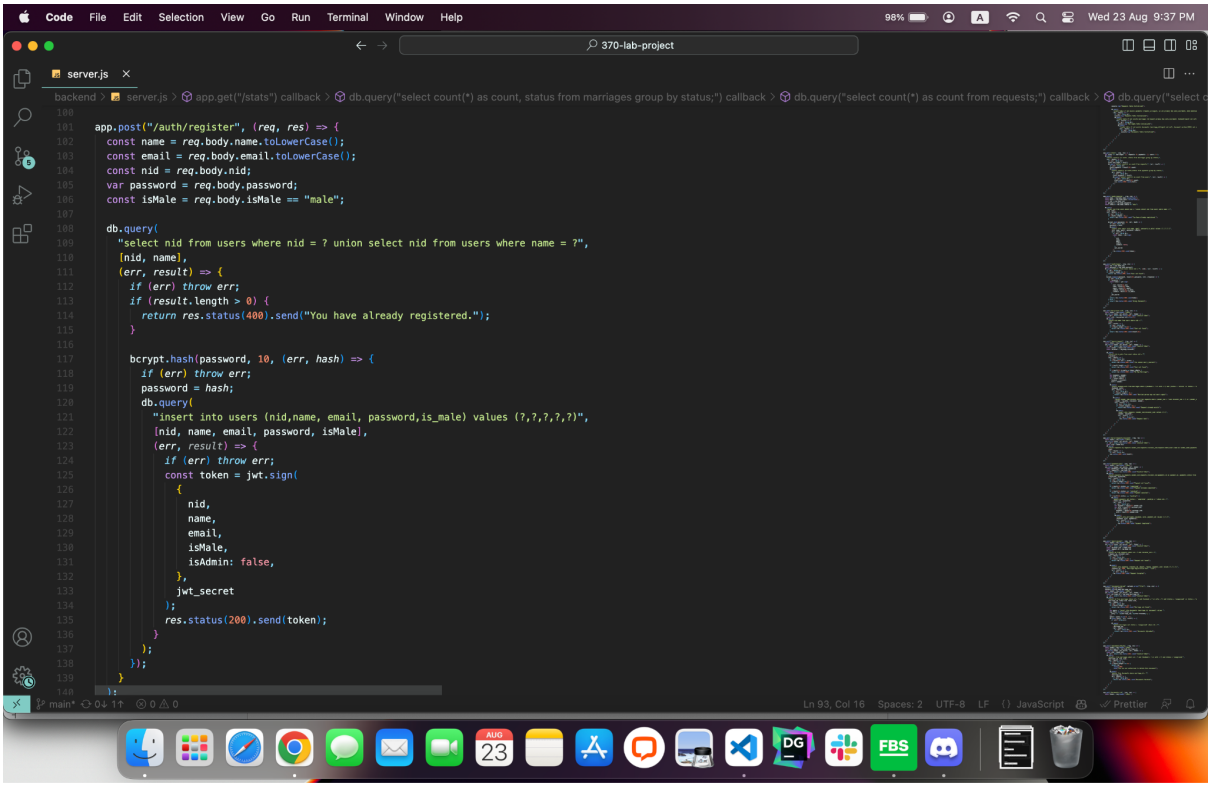
Tables are created if they do not exist the moment the backend startup. Other than basic processing, no extra libraries are used for mysql. Every set of data are queried with raw mysql which can be seen in the source code of the backend. The whole backend is written in a single file. Tables are created with proper foreign keys and datatypes.

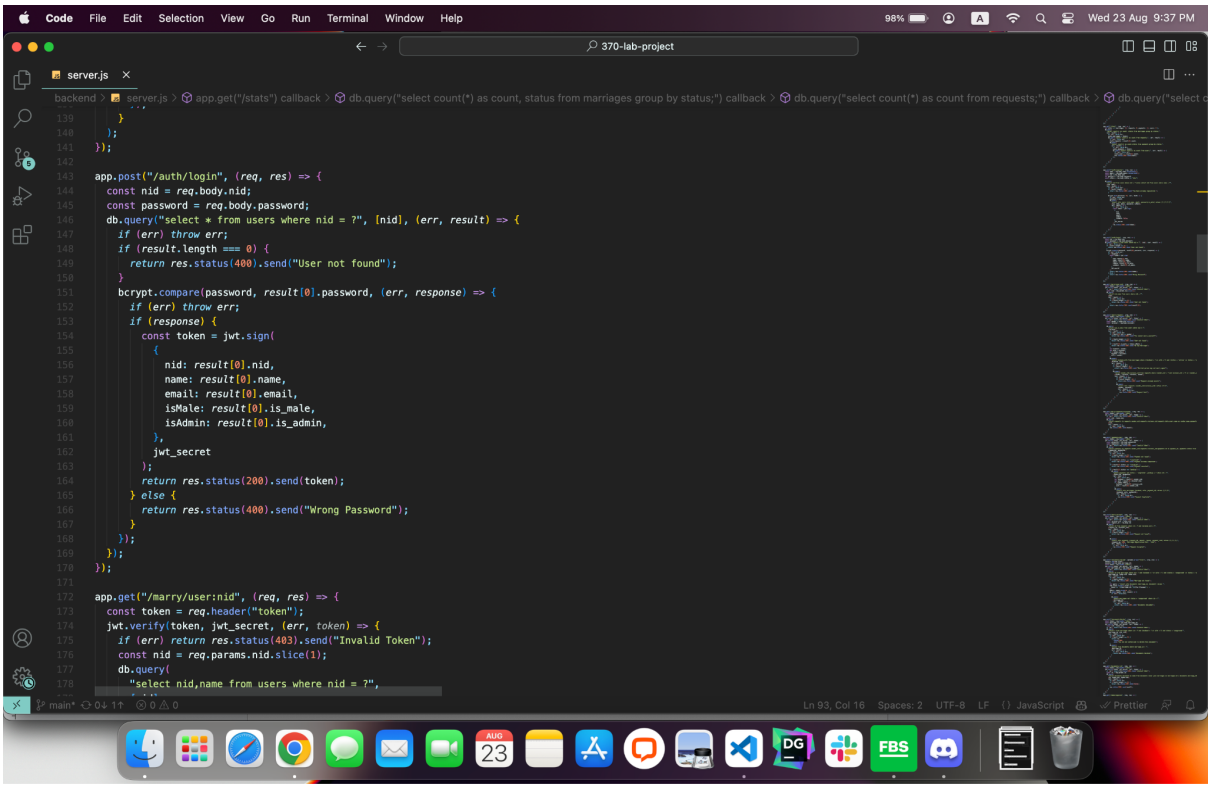
Here are some screenshots of the backend:

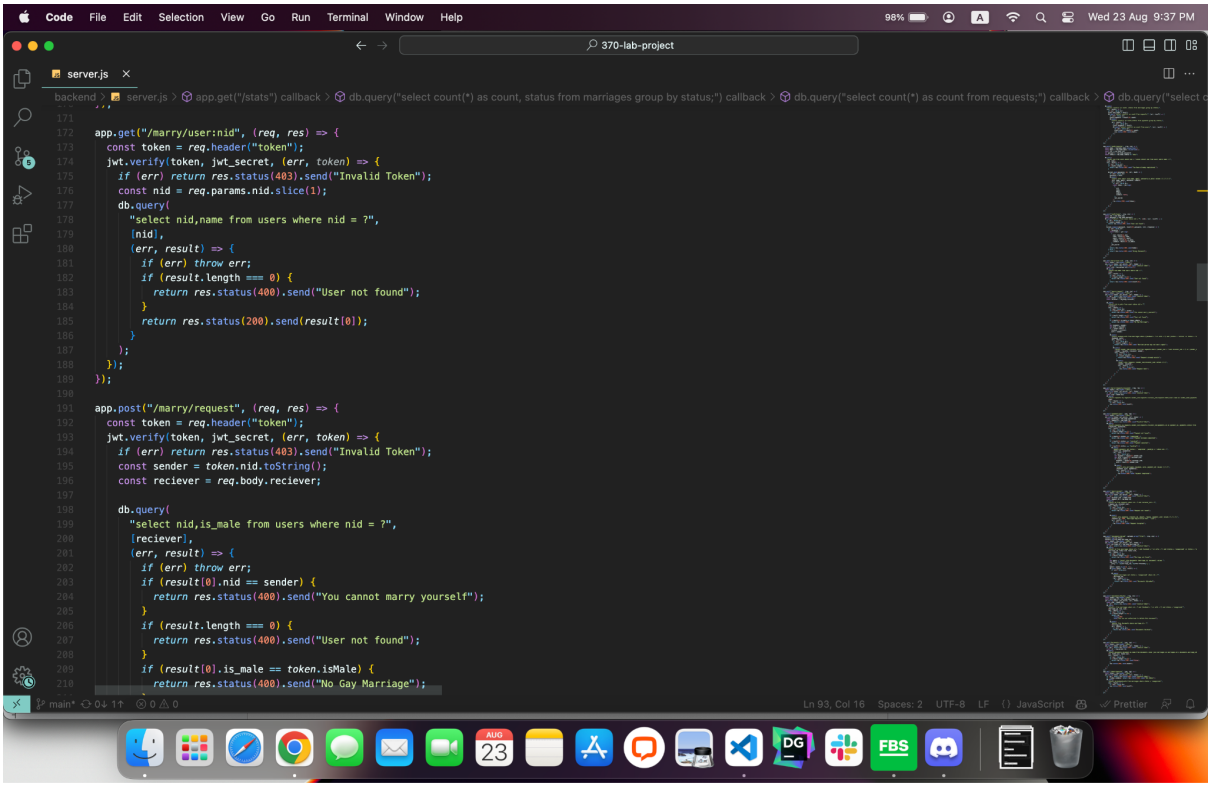


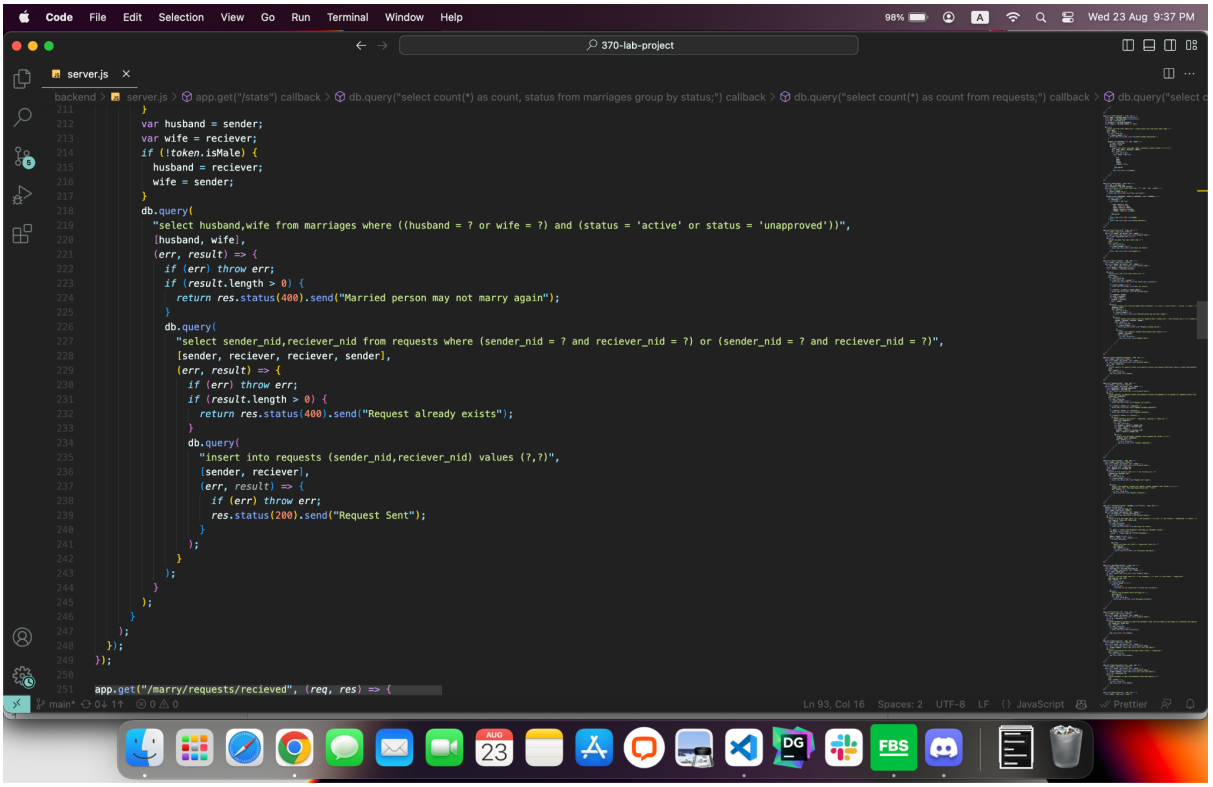

Code File Edit Selection View Go Run Terminal Window Help
Wed 23 Aug 9:37 PM
370-lab-project
server.js
backend > server.js > app.get("/stats") callback > db.query("select count(*) as count, status from marriages group by status;") callback > db.query("select count(*) as count from requests;") callback
db.connect((err) => {
if (err) throw err;
console.log("Connected to database");
db.query(
"create table if not exists users (nid bigint not null primary key, name varchar(255) not null, email varchar(255), password varchar(255) not null, is_admin boolean defa
(err, result) => {
if (err) throw err;
console.log("Users Table Initialized");
db.query(
"create table if not exists requests (id bigint primary key auto_increment, sender_nid bigint not null , reciever_nid bigint not null , date datetime default now(),
(err, result) => {
if (err) throw err;
console.log("Requests Table Initialized");
db.query(
"create table if not exists payments (request_id bigint, id int primary key auto_increment, date datetime default now(), status enum('pending','completed','cance
(err, result) => {
if (err) throw err;
console.log("Payments Table Initialized");
db.query(
"create table if not exists marriages (id bigint primary key auto_increment, husband bigint not null, wife bigint not null , date datetime DEFAULT now() , st
(err, result) => {
if (err) throw err;
console.log("Marriages Table Initialized");
db.query(
"create table if not exists documents (marriage_id bigint not null, document varchar(1024) not null, foreign key (marriage_id) references marriages(id));
(err, result) => {
if (err) throw err;
console.log("Documents Table Initialized");
}
);
}
);
}
);
}
);
}
);
});
Ln 93, Col 16 Spaces: 2 UTF-8 LF JavaScript Prettier

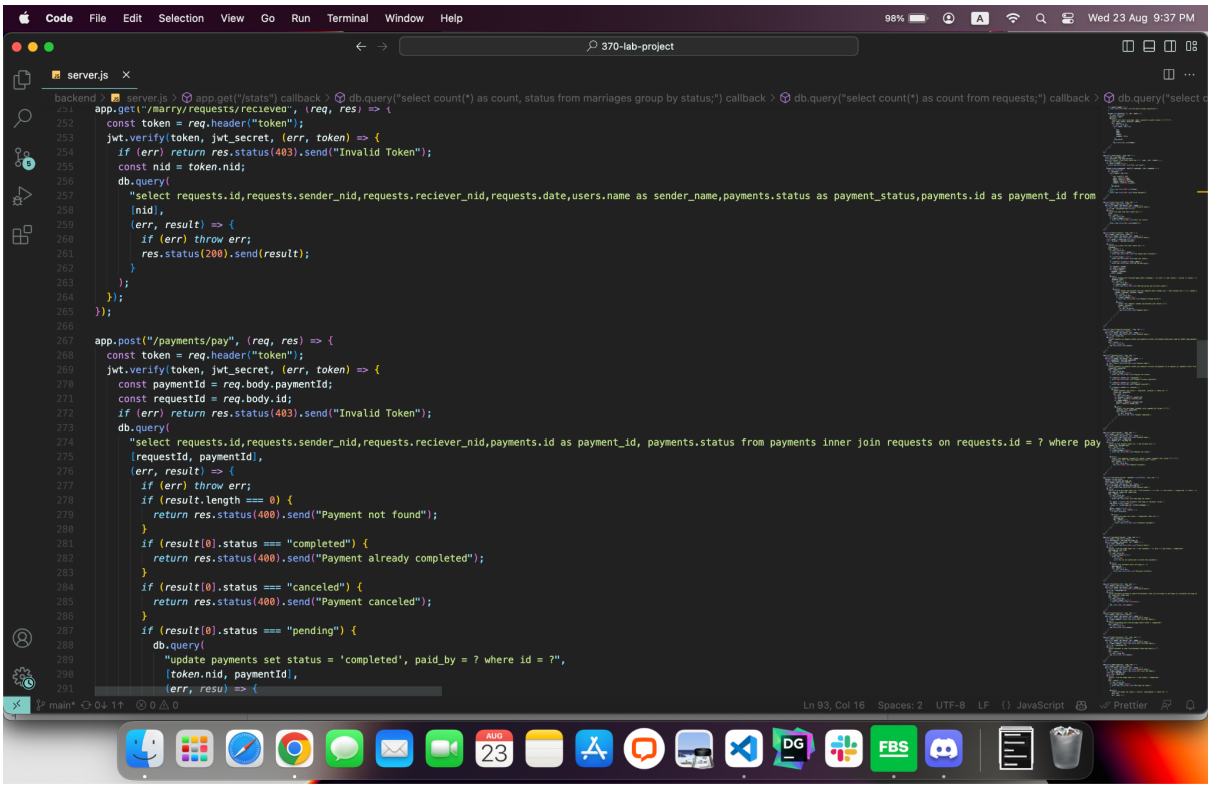

Code File Edit Selection View Go Run Terminal Window Help
Wed 23 Aug 9:37 PM
370-lab-project
server.js
backend > server.js > app.get("/stats") callback > db.query("select count(*) as count, status from marriages group by status;") callback > db.query("select count(*) as count from requests;") callback
app.get("/stats", (req, res) => {
var stats = { marriages: [], requests: 0, payments: [], users: 0 };
db.query(
"select count(*) as count, status from marriages group by status;",
(err, result) => {
if (err) throw err;
stats.marriages = result;
db.query("select count(*) as count from requests;", (err, result) => {
if (err) throw err;
stats.requests = result[0].count;
db.query(
"select count(*) as count,status from payments group by status;",
(err, result) => {
if (err) throw err;
stats.payments = result;
db.query("select count(*) as count from users;", (err, result) => {
if (err) throw err;
stats.users = result[0].count;
res.status(200).send(stats);
});
}
);
});
}
);
});
app.post("/auth/register", (req, res) => {
const name = req.body.name.toLowerCase();
const email = req.body.email.toLowerCase();
const nid = req.body.nid;
var password = req.body.password;
const isMale = req.body.isMale == "male";
db.query(
Ln 93, Col 16 Spaces: 2 UTF-8 LF JavaScript Prettier


Code File Edit Selection View Go Run Terminal Window Help
Wed 23 Aug 9:37 PM
370-lab-project
server.js
app.post("/auth/register", (req, res) => {
const name = req.body.name.toLowerCase();
const email = req.body.email.toLowerCase();
const nid = req.body.nid;
var password = req.body.password;
const isMale = req.body.isMale == "male";
db.query(
"select nid from users where nid = ? union select nid from users where name = ?",
[nid, name],
(err, result) => {
if (err) throw err;
if (result.length > 0) {
return res.status(400).send("You have already registered.");
}
bcrypt.hash(password, 10, (err, hash) => {
if (err) throw err;
password = hash;
db.query(
"insert into users (nid,name, email, password,is_male) values (?,?,?,?,?)",
[nid, name, email, password, isMale],
(err, result) => {
if (err) throw err;
const token = jwt.sign(
{
nid,
name,
email,
isMale,
isAdmin: false,
},
jwt_secret
);
res.status(200).send(token);
}
);
});
}
);


Code File Edit Selection View Go Run Terminal Window Help
Wed 23 Aug 9:37 PM
370-lab-project
server.js
app.post("/auth/login", (req, res) => {
const nid = req.body.nid;
const password = req.body.password;
db.query("select * from users where nid = ?", [nid], (err, result) => {
if (err) throw err;
if (result.length === 0) {
return res.status(400).send("User not found");
}
bcrypt.compare(password, result[0].password, (err, response) => {
if (err) throw err;
if (response) {
const token = jwt.sign(
{
nid: result[0].nid,
name: result[0].name,
email: result[0].email,
isMale: result[0].is_male,
isAdmin: result[0].is_admin,
},
jwt_secret
);
return res.status(200).send(token);
} else {
return res.status(400).send("Wrong Password");
}
});
});
});
app.get("/marry/user:nid", (req, res) => {
const token = req.header("token");
jwt.verify(token, jwt_secret, (err, token) => {
if (err) return res.status(403).send("Invalid Token");
const nid = req.params.nid.slice(1);
db.query(
"select nid,name from users where nid = ?",


Code File Edit Selection View Go Run Terminal Window Help
Wed 23 Aug 9:37 PM
370-lab-project
server.js
app.get("/marry/user:nid", (req, res) => {
const token = req.header("token");
jwt.verify(token, jwt_secret, (err, token) => {
if (err) return res.status(403).send("Invalid Token");
const nid = req.params.nid.slice(1);
db.query(
"select nid,name from users where nid = ?",
[nid],
(err, result) => {
if (err) throw err;
if (result.length === 0) {
return res.status(400).send("User not found");
}
return res.status(200).send(result[0]);
}
);
});
});
app.post("/marry/request", (req, res) => {
const token = req.header("token");
jwt.verify(token, jwt_secret, (err, token) => {
if (err) return res.status(403).send("Invalid Token");
const sender = token.nid.toString();
const reciever = req.body.reciever;
db.query(
"select nid,is_male from users where nid = ?",
[reciever],
(err, result) => {
if (err) throw err;
if (result[0].nid == sender) {
return res.status(400).send("You cannot marry yourself");
}
if (result.length === 0) {
return res.status(400).send("User not found");
}
if (result[0].is_male == token.isMale) {
return res.status(400).send("No Gay Marriage");


Code File Edit Selection View Go Run Terminal Window Help
Wed 23 Aug 9:37 PM
370-lab-project
server.js
backend > server.js > app.get("/stats") callback > db.query("select count(*) as count, status from marriages group by status;") callback > db.query("select count(*) as count from requests;") callback > db.query("select c
}
var husband = sender;
var wife = reciever;
if (!token.isMale) {
husband = reciever;
wife = sender;
}
db.query(
"select husband,wife from marriages where ((husband = ? or wife = ?) and (status = 'active' or status = 'unapproved'))",
[husband, wife],
(err, result) => {
if (err) throw err;
if (result.length > 0) {
return res.status(400).send("Married person may not marry again");
}
db.query(
"select sender_nid,reciever_nid from requests where (sender_nid = ? and reciever_nid = ?) or (sender_nid = ? and reciever_nid = ?)",
[sender, reciever, reciever, sender],
(err, result) => {
if (err) throw err;
if (result.length > 0) {
return res.status(400).send("Request already exists");
}
db.query(
"insert into requests (sender_nid,reciever_nid) values (?,?)",
[sender, reciever],
(err, result) => {
if (err) throw err;
res.status(200).send("Request Sent");
}
);
}
);
}
);
}
);
});
});
app.get("/marry/requests/recieved", (req, res) => {
main* 0↓ 1↑ 0 0
Ln 93, Col 16 Spaces: 2 UTF-8 LF {} JavaScript Prettier


Code File Edit Selection View Go Run Terminal Window Help
Wed 23 Aug 9:37 PM
370-lab-project
server.js
backend > server.js > app.get("/stats") callback > db.query("select count(*) as count, status from marriages group by status;") callback > db.query("select count(*) as count from requests;") callback > db.query("select c
app.get("/marry/requests/recieved", (req, res) => {
const token = req.header("token");
jwt.verify(token, jwt_secret, (err, token) => {
if (err) return res.status(403).send("Invalid Token");
const nid = token.nid;
db.query(
"select requests.id,requests.sender_nid,requests.reciever_nid,requests.date,users.name as sender_name,payments.status as payment_status,payments.id as payment_id from
[nid],
(err, result) => {
if (err) throw err;
res.status(200).send(result);
}
);
});
});
app.post("/payments/pay", (req, res) => {
const token = req.header("token");
jwt.verify(token, jwt_secret, (err, token) => {
const paymentId = req.body.paymentId;
const requestId = req.body.id;
if (err) return res.status(403).send("Invalid Token");
db.query(
"select requests.id,requests.sender_nid,requests.reciever_nid,payments.id as payment_id, payments.status from payments inner join requests on requests.id = ? where pay
[requestId, paymentId],
(err, result) => {
if (err) throw err;
if (result.length === 0) {
return res.status(400).send("Payment not found");
}
if (result[0].status === "completed") {
return res.status(400).send("Payment already completed");
}
if (result[0].status === "canceled") {
return res.status(400).send("Payment canceled");
}
if (result[0].status === "pending") {
db.query(
"update payments set status = 'completed', paid_by = ? where id = ?",
[token.nid, paymentId],
(err, resu) => {
main* 0↓ 1↑ 0 0
Ln 93, Col 16 Spaces: 2 UTF-8 LF {} JavaScript Prettier

Conclusion

In wrapping up this project, I've gained a ton of valuable insights into the world of web development and technology integration. Working on the marriage registration portal using Node.js, React.js, Express.js, and MySQL has been an incredible learning journey for me.

Dealing with data using MySQL was a big learning curve. Designing the database schema, setting up relationships between tables, and writing queries challenged me to think about data organization and retrieval in a more structured way. I've gained hands-on experience in database management and ensuring data integrity, which are skills that I believe will serve me well in future projects.

References

1. https://chakra-ui.com/getting-started
2. https://expressjs.com/en/starter/hello-world.html
3. https://react.dev/