```
const express = require('express');
const { body, validationResult } = require('express-validator'); // Import from express-validator
const bodyParser = require('body-parser');
const app = express();
const pool = require('./db'); // Assuming you have a file for DB connection
const PORT = 3000;
// Middleware to parse JSON
app.use(express.json());

// CORS Middleware (manual)
app.use((req, res, next) => {
   res.header("Access-Control-Allow-Origin", "http://localhost:5173"); // Allow requests from this origin
   res.header("Access-Control-Allow-Methods", "GET,HEAD,PUT,PATCH,POST,DELETE"); // Allow methods
   res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept"); // Allow headers
   if (req.method === "OPTIONS") {
      return res.sendStatus(204); // Respond with 204 for preflight requests
   }
   next(); // Move to the next middleware or route handler
});

/*-------------------------------------------------------------------------*/
//Food Group APIs
// Route to get all food groups
app.get('/food_group', async (req, res) => {
try {
const result = await pool.query('SELECT * FROM food_group');
res.status(200).json({ status: "200", foodgroup: result.rows });
} catch (err) {
console.error('Error fetching data:', err.message);
res.status(500).send('Server Error');
}
});
```

```
app.get('/foodgroupid',[
   body('id').notEmpty().withMessage('id is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {id}=req.body;
      const rs =  await pool.query('select * from food_group where gid=$1',[id]);


      if(rs.rows.length>0){


         res.send({status:"200",message:"Success",data:result.rows})
      } else{
         res.send({status:"400",message:"No DATA Found"})


      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');


   }
});
```

```
app.delete('/delfoodgroup',[
   body('id').notEmpty().withMessage('id is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {id}=req.body;
      const rs =  await pool.query('select * from food_group where gid=$1',[id]);

      if(rs.rows.length>0){
         await pool.query('delete from food_group where gid=$1',[id]);
         res.send('{status:"200",message:"Delete Success"}')
      } else{
         res.send('{status:"400",message:"Delete Failed"}')

      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');

   }
});
```

```
app.put('/updatefoodgroup',[
   body('group_name').notEmpty().withMessage('group_name is required'),
   body('gid').notEmpty().withMessage('gid is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {group_name,gid}=req.body;
      const rs =  await pool.query('select * from food_group where gid=$1',[id]);

      if(rs.rows.length>0){
         await pool.query('update food_group set group_name=$1 where gid=$2',[group_name,gid]);
         res.send('{status:"200",message:"Update Success"}')
      } else{
         res.send('{status:"400",message:"Update Failed"}')

      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');

   }
});
```

```
app.put('/addfoodgroup',[
   body('group_name').notEmpty().withMessage('group_name is required'),
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {group_name}=req.body;
      const rs =  await pool.query('INSERT INTO food_group(group_name) VALUES($1) RETURNING * ',[group_name]);
      res.send('{status:"200",message:"Food Group Save Successfuly"}')
      }
   } catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');

   }
});
/*-------------------------------------------------------------------------------*/
//QTY APIs
app.get('/qtymast', async (req, res) => {
   try {
      const result = await pool.query('SELECT * FROM qtymast');
      res.status(200).json({ status: "200", qtymast: result.rows });
   } catch (err) {
      console.error('Error fetching data:', err.message);
      res.status(500).send('Server Error');
   }
});
```

```
app.get('/qtymastid',[
   body('qid').notEmpty().withMessage('qid is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {qid}=req.body;
      const rs =  await pool.query('select * from qtymast where qid=$1',[qid]);


      if(rs.rows.length>0){


         res.send({status:"200",message:"Success",data:rs.rows})
      } else{
         res.send({status:"400",message:"No DATA Found"})


      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');


   }
});
```

```
app.delete('/delqtymast',[
   body('qid').notEmpty().withMessage('qid is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {qid}=req.body;
      const rs =  await pool.query('select * from qtymast where qid=$1',[qid]);

      if(rs.rows.length>0){
         await pool.query('delete from qtymast where qid=$1',[qid]);
         res.send('{status:"200",message:"qtymast Delete Success"}')
      } else{
         res.send('{status:"400",message:"qtymast Delete Failed"}')

      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');

   }
});
```

```
app.put('/updateqtymast',[
   body('qty_type').notEmpty().withMessage('qty_type is required'),
   body('qid').notEmpty().withMessage('qid is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {qty_type,qid}=req.body;
      const rs =  await pool.query('select * from qtymast where qid=$1',[qid]);

      if(rs.rows.length>0){
         await pool.query('update qtymast set qty_type=$1 where qid=$2',[qty_type,qid]);
         res.send('{status:"200",message:"Update Success"}')
      } else{
         res.send('{status:"400",message:"Update Failed"}')

      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');
   }
});
```

```
app.put('/addqtymast',[
   body('qty_type').notEmpty().withMessage('qty_type is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {qty_type}=req.body;
      const rs =  await pool.query('INSERT INTO qtymast(qty_type) VALUES($1) RETURNING * ',[qty_type]);
      res.send('{status:"200",message:"Qtymast Save Successfuly"}')
      }
   } catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');

   }
});

/*-------------------------------------------------------------------------------*/
//Menu APIs

app.get('/menu', async (req, res) => {
   try {
      const result = await pool.query('SELECT * FROM menu');
      res.status(200).json({ status: "200", menu: result.rows });
   } catch (err) {
      console.error('Error fetching data:', err.message);
      res.status(500).send('Server Error');
   }
});
```

```
app.get('/menubyid',[
   body('id').notEmpty().withMessage('id is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {id}=req.body;
      const rs =  await pool.query('select * from menu where mid=$1',[id]);


      if(rs.rows.length>0){


         res.send({status:"200",message:"Success",data:rs.rows})
      } else{
         res.send({status:"400",message:"No DATA Found"})


      }
   }


   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');


   }
});
```

```
app.delete('/delmenu',[
   body('mid').notEmpty().withMessage('mid is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {mid}=req.body;
      const rs =  await pool.query('select * from menu where mid=$1',[mid]);

      if(rs.rows.length>0){
         await pool.query('delete from menu where mid=$1',[mid]);
         res.send('{status:"200",message:"menu Delete Success"}')
      } else{
         res.send('{status:"400",message:"menu Delete Failed"}')

      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');

   }
});
```

```
app.put('/updatemenu',[
   body('menu_name').notEmpty().withMessage('menu_name is required'),
   body('menu_price').notEmpty().withMessage('menu_price is required'),
   body('gid').notEmpty().withMessage('gid is required'),
   body('qid').notEmpty().withMessage('qid is required'),
   body('mid').notEmpty().withMessage('mid is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {menu_name,menu_price,gid,qid,mid}=req.body;
      const rs =  await pool.query('select * from menu where mid=$1',[mid]);

      if(rs.rows.length>0){
         await pool.query('update menu set
menu_name=$1,menu_price=$2,gid=$3,qid=$4,mid=$5',[menu_name,menu_price,gid,qid,mid]);
         res.send('{status:"200",message:"Update Success"}')
      } else{
         res.send('{status:"400",message:"Update Failed"}')

      }
   }

   }catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');

   }
});
```

```
app.put('/addmenu',[
   body('menu_name').notEmpty().withMessage('menu_name is required'),
   body('menu_price').notEmpty().withMessage('menu_price is required'),
   body('gid').notEmpty().withMessage('gid is required'),
   body('qid').notEmpty().withMessage('qid is required')
],async(req,res) =>{
   try{
      const errors=validationResult(req);
   if (!errors.isEmpty()){
      return res.status(400).json({errors:errors.array()});
   }else {
      const {menu_name,menu_price,gid,qid}=req.body;
      const rs =  await pool.query('INSERT INTO menu(menu_name,menu_price,gid,qid) VALUES($1,$2,$3,$4)
RETURNING * ',[menu_name,menu_price,gid,qid]);
      res.send('{status:"200",message:"Menu Save Successfuly"}')
      }
   } catch(err) {
      console.error(err.message);
      res.status(500).send('Server error');
  } });
 /*-------------------------------------------------------------------------------*/
 //Menu CARD API Visible to Customer
app.get('/menucard', async(req,res)=>{
try{
const result=await pool.query('SELECT menu_name,menu_price,group_name,qty_type FROM
menu,food_group,qtymast WHERE food_group.gid=menu.gid AND menu.qid=qtymast.qid')
res.json({status:"200",menucard:result.rows});
}catch(err) {
console.error(err.message);
res.status(500).send('Server error');
}
});
```

```
/*-------------------------------------------------------------------------------*/

// Start the server

app.listen(PORT, () => {

console.log(`Server is running on port http://localhost:${PORT}`);


});
```