MIDTERM PROJECT

RESTAURANT DATABASE WEBSITE

111110544 LIHEWEI

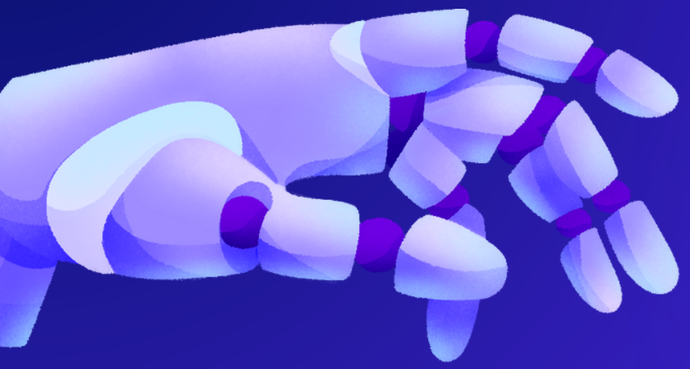
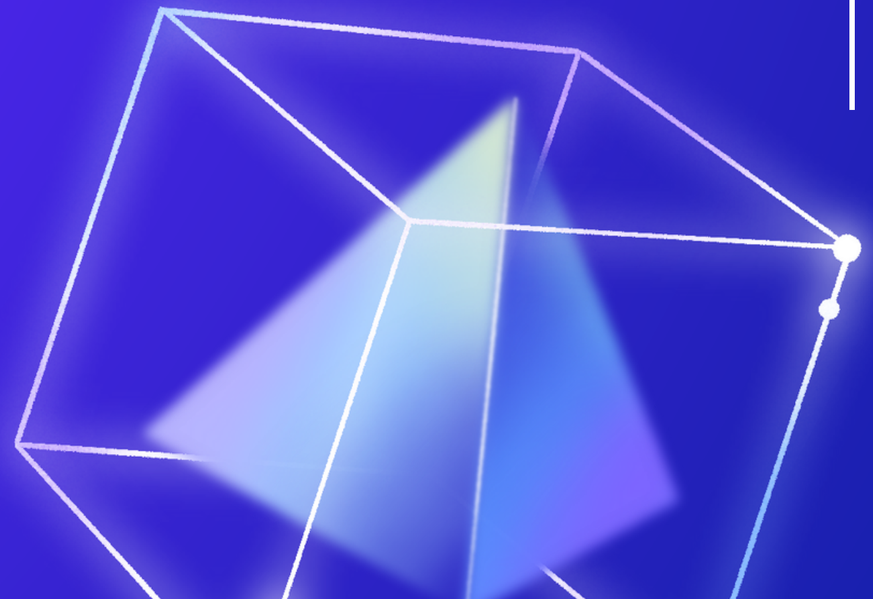


TABLE OF CONTENTS

- Introduction 01
- Website looks like 02
- Project Structure 03
- Database 04
- Server-Side Code 05
- Client-Side Code 06
- Challenges and Solutions 07



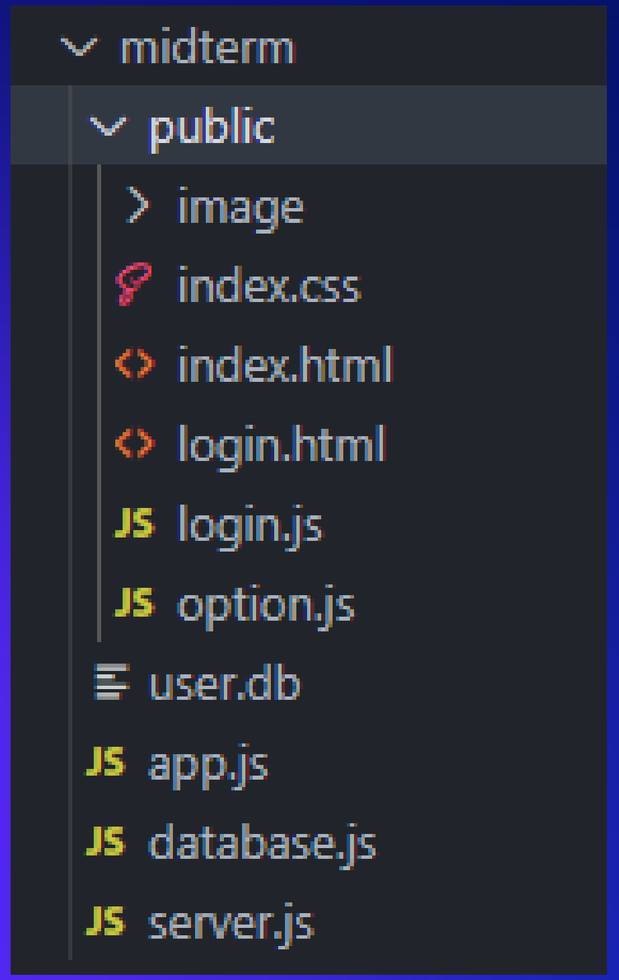
INTRODUCTION

Presenting my 'Restaurant Database Website'

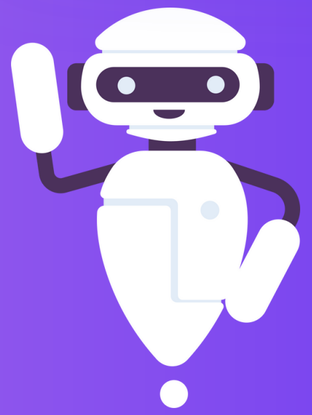
Platform with secure login for admins and user-friendly online ordering.
Explore key features and tech stack

WEBSITE LOOKS LIKE

PROJECT STRUCTURE



DATABASE



user.db

midterm > JS database.js > ...

```
1  import { DB } from "https://deno.land/x/sqlite/mod.ts";
2
3  let db = null
4  export async function open() {
5      db = new DB("user.db");
6      db.query(`CREATE TABLE IF NOT EXISTS users
7      (uid INTEGER PRIMARY KEY AUTOINCREMENT,
8       user TEXT, pass TEXT, email TEXT)`)
9  }
10
11 export async function userAdd(user) {
12     db.query(`INSERT INTO users (user, pass, email) VALUES (?,?,?)`, [user.user,
       user.pass, user.email])
13   }
14
15 export async function userGet(user1) {
16   let q = db.query(`SELECT uid, user, pass, email FROM users WHERE user=?`, [user1])
17   console.log(`userGet(${user1})=${q}`)
18   if (q.length <=0) return null
19   let [uid, user, pass, email] = q[0]
20   return {uid, user, pass, email}
21 }
```

SERVER-SIDE CODE

```
import {Server, sendJson, bodyParams, sendStatus, Status} from './server.js'
import * as db from './database.js'

db.open()

const server = new Server()
server.public("/public")

server.router.get('/', home)
server.router.post('/login', login)
server.router.post('/signup', signup)

async function home(ctx) {
    ctx.response.redirect("/public/#login")
}

async function signup(ctx) {
  const params = await bodyParams(ctx)
  console.log('params=', params)
  let user = await db.userGet(params.user)
  if (user == null) {
    console.log('signup:params=', params)
    await db.userAdd({user:params.user, pass:params.password, email:params.email})
    sendStatus(ctx, Status.OK)
  }
  else
    sendStatus(ctx, Status.Fail)
}

```

```
async function login(ctx) {
  const params = await bodyParams(ctx)
  let user = await db.userGet(params.user)
  console.log('login:user=', user)
  if (user != null && user.pass == params.password) {
    await ctx.state.session.set('user', user)
    sendStatus(ctx, Status.OK)
  } else
    sendStatus(ctx, Status.Fail)
}

await server.listen(8000)

```

APP.JS

SERVER-SIDE CODE

```
import { Application, Router, send } from "https://deno.land/x/oak/mod.ts";
import { Session } from "https://deno.land/x/oak_sessions/mod.ts";

export class Server {
    constructor() {
        this.app = new Application()
        this.router = new Router()
    }
    public(path) {
        this.router.get(`${path}/(.*)`, async (ctx)=>{
          console.log(ctx.request.url.pathname)
          await send(ctx, ctx.request.url.pathname, {
              root: `${Deno.cwd()}/`,
              index: "login.html",
          })
        })
    }
    async listen(port) {
        this.app.use(Session.initMiddleware())
        this.app.use(this.router.routes())
        this.app.use(this.router.allowedMethods())
        console.log(`Server run at http://127.0.0.1:${port}`)
        await this.app.listen({ port })
    }
}

export function sendJson(ctx, obj) {
  ctx.response.type = 'application/json'
  ctx.response.body = obj
}

```

```
export const Status = {
  OK:200,
  Fail:400,
  Unauthorized:401,
  Forbidden:403,
  NotFound:404,
}

export function sendStatus(ctx, status) {
  ctx.response.type = 'application/json'
  ctx.response.status = status
  ctx.response.body = {status}
}

export let body: any bodyParams(ctx) {
  let body = ctx.request.body()
  if (body.type === "json") {
    let params = await body.value
    console.log('bodyParams:', params)
    return params
  } else {
    console.log(`Error: body.type=${body.type}`)
  }
}
```

SERVER.JS

CLIENT-SIDE CODE

THANK YOU!