Autoservis dokumentace

Adam Pietrosch – 3.B

Obsah

Technická část

Základní technické informace:

- Backendový framework = Express
- Primární programovací jazyk = Javascript
- Databáze = Mongo

Použité knihovny a doplňky:

- Mongoose (k práci s databázovými modely)
- Express-session (ke správě serverových sessions)
- Body-parser (ke zpracování http požadavků v různých formátech)
- Cookie-parser (ke správě cookies)
- Passport (Na autentizaci uživatelů)
- Connect-mongo (pro ukládání serverových sessions byla použita mongo databáze)
- Ejs (view engine, se kterým vytváříme dynamické html stránky)
- Bcrypt (knihovna na hashování)

Struktura projektu:

- Použit model MVC, jsou zde tři hlavní složky: models (obsahuje databázové modely), routes (ke správě url a http požadavků) a views (obsahuje ejs soubory s html kódem)

Příklad z routeru:

- Pokud se pošle autorizovaný http POST požadavek na url „/employees-add“ s daty: jméno a číslo zaměstnance, tak se vytvoří nový zaměstnanec s danými daty a uloží se do databáze a poté server přesměruje uživatele na seznam všech zaměstnanců

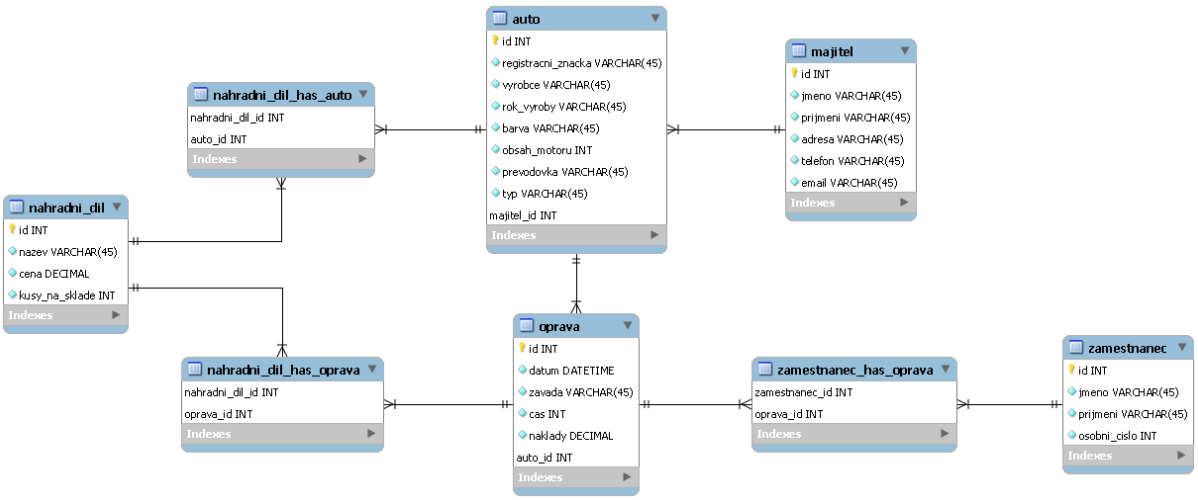
```
23  router.post('/employees-add', async (req, res) => {
24
25      const employee = new Employee({
26          name: req.body.name,
27          personal_number: req.body.personal_number
28      })
29
30      await employee.save()
31      res.redirect('/admin/employees')
32  })
33
```

Příklad autentizační metody:

- V odeslaném http POST požadavku k přihlášení uživatele nesmí chybět data email a password, jinak vrátí metoda chybovou hlášku
- Metoda první ověří, jestli existuje uživatel s daným emailem a poté ověří, jestli se heslo shoduje s heslem v databázi
- V případě špatných údajů vrátní metoda uživateli chybovou hlášku

```
38  // LOGIN USER
39  passport.use('local.login', new localStrategy({
40      usernameField: 'email',
41      passwordField: 'password',
42      passReqToCallback: true
43  },
44  async (req, email, password, done) => {
45
46      req.flash('postedData', req.body)
47
48      const user = await User.findOne({ email: email })
49      if (user == null) {
50          return done(null, false, { message: 'Špatný email.' })
51      }
52
53      const isPassValid = await bcrypt.compare(password, user.password)
54      if (!isPassValid) {
55          return done(null, false, { message: 'Špatné heslo.' })
56      }
57
58      return done(null, user)
59  }
60  ))
```

Model databáze:



Uživatelská část

Zprovoznění systému:

1. Zkopírovat všechny soubory, např. z Githubu
2. Nainstalovat nejnovější verzi Node.js (pokud ho už na počítači nemáte)
3. Otevřít si příkazový řádek s umístěním v adresáři projektu
4. Napište a spusťte příkaz: ,,npm install“ (nainstalují se všechny potřebné balíčky)
5. V Kořenovém adresáři projektu vytvořte soubor, který pojmenujte: .env a zde musíte napsat na řádek: DATABASE= a za to url k vaší mongo databázi
6. Server spustíte příkazem: ,,npm run dev“

Ukázkový web:

- Není k dispozici
- Nemám na tento projekt už volné zdroje
- Možná ukázka na vyžádání

Popis webu:

- Funkční přihlašování a registrace
- Po přihlášení můžete spravovat: zákazníky, zaměstnance, a opravy v přehledných tabulkách

Závěr:

Projekt má základní funkce, které má mít a server je plně funkční. Slabší část je frontend, na kterém by byla potřeba udělat ještě hodně práce, ale neměl jsem na to čas.