

Projekt Mini-Axami

Skapa ett system för husägare. (Users i ett vanligt system)

Ägare av produkten skall kunna göra följande:

- Skapa kund som skall få inloggning till systemet. (create)
- Ta bort kund som har inloggning till systemet. (delete)
- Editera uppgifter om kund som har inloggning till systemet. (edit/update)
- Lista och söka efter kunder inför delete eller update (read)

Kund i produkten skall kunna göra följande

- Lägga upp ett nytt hus/fastighet (Create)
- Lägga till statisk info om hus/fastighet (edit/update/create)
- Ta bort hus/fastighet (delete)
- Lägga till boende/hantverkare/användare (BHA) per hus (create)
- Lägga till uppgift för BHA
 - Uppgiften skall ha antingen hög eller låg prioritet. Uppgiften skall ha deadline
 - Uppgiften skall antingen tilldelas BHA eller vara öppen för vem som helst att ta sig an.
 - Om uppgift tilldelas BHA som finns i systemet skall denne få mail om uppgiften.

Read...

BHA i produkten skall kunna göra följande

- Ta sig an en uppgift: Då skall följande skrivas in:
 - Ungefärlig tid för genomförande
 - Ungefärlig kostnad (default 0 kr)
 - Övrig kommentar
- Mail skall gå till husägare med info om att uppgiften är antagen.
- Klarmarkera en uppgift – med möjlighet att uppdatera tid, kostnad och kommentarer.
- Säga upp sig från uppgift – Mail går till husägare med relevant info.

Projektplan skall utöver ovanstående innehålla:

Val av:

- Databas (mysql, mongodb, repldb)
- serverspråk/miljö
- eventuella ramverk på både serversidan och frontend
- typ av applikation: SPA, SSR eller hybrid
- autentiseringsmetod password eller passwordless
- auktoriseringsmetod : JWT eller sessions
- Css-ramverk
- PWA

Rekommendationer för enklaste setup

PHP + Mysql + SSR + Passwords + SESSION

Node/Express + replDb + SSR + Passwords + SESSION

Rekommendationer för modulär setup med möjlighet att koppla på andra tekniker såsom native app, desktop-app.

PHP + MYSQL + REST-api + Passwordless + JWT

Node/Express + MYSQL/mongoDb + REST-api + Passwordless + JWT

TIPS

Oavsett setup ni väljer är det bra att börja med en databas.

Fundera ordentligt vilka relationer och kopplingar ni behöver.

Börja sedan med serverkod och testa den med postman eller liknande innan ni skapar användargränssnitt.

Användargränssnitt brukar nämligen göra att man fastnar i CSS och front-end-problematik i onödan.

Om allt som har med servern är klart är det enklare att bygga upp sitt användargränssnitt.