

Boiler Room vecka 6-8: Grupprojekt: Reseplaneraren 4.0

Tema: Grupprojektet handlar om att vidareutveckla **Reseplaneraren**, en applikation som nu ska bli en fullständig, flersidig och optimerad React-applikation. Nya ämen för dessa veckor är: global state-hantering och samarbete i utvecklingsteam.

Projektbeskrivning: Flersidig och optimerad Reseplanerare v4.0

Beskrivning: Bygg vidare på er tidigare applikation genom att implementera tidigare ämnen samt nya ämnet: global state-hantering med Redux Toolkit.

Exempel på funktionalitet:

- Dynamisk navigering mellan olika vyer:
 - Hem (Översikt över planerade resor).
 - Skapa ny resa (Formulär för att lägga till resor).
 - Detallsida (Visar mer information om en vald resa).
 - Global state management för att hantera resedata centralt.
 - State management med useState och useEffect för att hantera dynamiska uppdateringar och effekter i komponenter.
 - Prestandaoptimering genom memoization och lazy loading.
 - Tidigare funktionalitet från Boiler Room-dokumentationen ska finnas kvar och integreras i den slutliga applikationen.
-

Mål för projektet

1. **Kunskapsmål:**
 - K1: Förklara hur Reacts komponentbaserade arkitektur möjliggör återanvändbarhet och effektiv utveckling.
 - K2: Beskriva hur state management hanteras i React med hooks som useState och useEffect.
 - K3: Förklara hur React Router används för att skapa dynamisk navigering i applikationer.
 - K4: Beskriva tekniker för att optimera prestanda i React-applikationer (memoization, lazy loading).
 - K5: Förklara hur global state hanteras med hjälp av Redux Toolkit.
 - K10: Redogöra för centrala principer för god dokumentation.
2. **Färdighetsmål:**
 - F1: Skapa och hantera modulära komponenter i React.
 - F2: Implementera state management med hooks och global state-lösningar.
 - F4: Optimera prestanda i React-applikationer.

3. Förbättra samarbetsförmåga i utvecklingsteam:

- Versionshantering och Git/GitHub-flöden
 - Kodstandarder och kodgranskning
 - Dokumentation och teamkommunikation
-

Krav för projektet

1. Applikationsstruktur och Routing (K3)

- Implementera en **flersidig React-applikation** med **React Router**.
- Minst tre vyer:
 - **Hem** (Översikt över resor)
 - **Lägg till ny resa** (Formulär för att skapa en resa)
 - **Detaljsida** (Visar detaljer om en specifik resa)
- Implementera **useParams** för att hämta dynamiska parametrar i rutter.

2. State Management med Hooks och Global State:

- Använd **useState** för att hantera lokalt state i enskilda komponenter.
- Använd **useEffect** för att hantera sidoeffekter, t.ex. datahämtning eller uppdateringar.
- Implementera **Redux Toolkit** för att hantera resedata centralt.
- Implementera **actions** och **reducers** för att lägga till, uppdatera och ta bort resor.

3. Prestandaoptimering (K4, F4)

- **Memoization** av komponenter och data med **React.memo** eller **useMemo**.
- **Lazy loading** av komponenter med **React.lazy** och **Suspense**.
- Optimera rendering av listor och dataflöden.

5. Dokumentation och Kodkvalitet (K10)

- **README-fil** som beskriver projektets struktur och hur man kör applikationen.
 - Kodkommentarer och förklaringar vid behov.
 - **Versionshantering med Git/GitHub:**
 - Feature branches och pull requests
 - Minst 3 commits per utvecklare
-

Level Ups (Frivilliga Utmaningar)

- Implementera **autentisering** (t.ex. inloggning med Firebase/Auth0).
- Lägg till **API-integration** för att hämta platser eller väderdata.
- Implementera **offline-stöd** (PWA eller LocalStorage).
- Designa en **dashboardvy** för att se statistik över resor.

- Implementera **push-notifikationer** eller mailfunktion vid skapade resor.
-

OBS: Detta är ett grupprojekt.

Arbetsprocess tips:

Sprint 1: Planering och Setup

- Skapa en backlog och definiera **Minimum Viable Product (MVP)**.
- Fördela uppgifter och bestäm roller inom gruppen.
- Sätta upp GitHub-repository och versionshanteringsflöde.

Sprint 2: Grundläggande funktionalitet

- Implementera routing, state management, global state management och grundläggande UI.
- Testa flöden och iterationer.

Sprint 3: Optimering och slutgiltiga funktioner

- Prestandaoptimeringar och kodförbättringar.
- Implementering av Level Ups om tid finns.

Sprint 4: Finslipning och Inlämning

- Dokumentation och kodgranskning.
 - Förberedelse för presentation och inlämning.
-

Presentation och Inlämning

- **Deadline:** Vecka 8, torsdag
 - **Presentation i grupp:** Vecka 8, fredag
 - **Innehåll i presentationen:**
 - Demonstration av applikationen.
 - Genomgång av kodstruktur och samarbetsprocess.
 - Reflektion över utmaningar och lärdomar.
-

Sammanfattning

Detta gruppprojekt syftar till att bygga en avancerad React-applikation genom att kombinera tidigare lärda koncept från Boiler Room-dokumentent. Studenterna får öva på att bygga

skalbara lösningar, hantera global state och optimera prestanda samtidigt som de samarbetar effektivt i utvecklingsteam.

Lycka till med projektet!