**Viikko 1: Projektin Alkuvalmistelut ja Käyttöliittymän Perusta**

**Projektin Perustaminen:**

* GitHub-repositorion luominen.
* README.md-tiedoston alustus (projektin kuvaus, asennusohjeet).
* .gitignore-tiedoston luominen.

**Kehitysympäristön Pystytys:**

* Tarvittavien ohjelmistojen ja kirjastojen asennus (esim. Node.js, npm/yarn, mahd. UI-framework).
* Projektin alustaminen valitulla frameworkilla (esim. create-react-app, vue create).
* Ympäristömuuttujien määrittely (jos tarpeen).

**Yhteiset Päätökset ja Käytännöt:**

* Päätös käytettävistä kirjastoista (esim. local storage -kirjasto, UI-komponenttikirjasto, kaaviokirjasto).
* Koodauskäytäntöjen sopiminen (nimeäminen, kommentointi, tiedostorakenne).
* Versionhallintakäytäntöjen sopiminen (branching-malli, pull request -prosessi).
* Kommunikaatiokäytäntöjen sopiminen (palaverit, viestintäkanavat).

**Käyttöliittymän Perusrakenne:**

* Pääasettelun (Layout) luominen (header, footer, pääsisältöalue).
* Responsiivisuuden varmistaminen.
* Navigointipalkin tai -valikon toteutus.
* Linkkien luominen eri osioihin (palkanlaskenta, budjetti, raportit).
* Perustyylien määrittely (väripaletti, fontit, typografia).
* Mahdollisen CSS-preprosessorin käyttöönotto.
* Yleisten tyyliluokkien luominen (esim. painikkeet, lomake-elementit).

**Viikko 2: Palkanlaskenta**

**Palkkatietojen Malli ja Tallennus (Local Storage):**

* Funktioiden luominen oman palkkatiedon (palkkatyyppi, tuntimäärä/kuukausipalkka, veroprosentti) tallentamiseen, hakemiseen, päivittämiseen ja poistamiseen local storagessa.
* Selkeiden avainten määrittely local storageen (esim. mySalary).

**Palkkatietojen Käyttöliittymä:**

* Lomakkeen luominen omien palkkatietojen syöttämiseen ja muokkaamiseen.
* Tietojen synkronointi local storagen kanssa.

**Bruttopalkan Laskenta:**

* Funktion luominen bruttopalkan laskemiseen eri palkkatyyppien perusteella.
* Yksikkötestien kirjoittaminen bruttopalkan laskennalle.

**Verotuksen ja Vähennysten Laskenta:**

* Funktioiden luominen *yksinkertaistettujen* verojen ja pakollisten vähennysten laskemiseen.
* Yksikkötestien kirjoittaminen verojen ja vähennysten laskennalle.

**Palkkalaskelman Luonti ja Näyttäminen:**

* Funktion luominen palkkalaskelman tietojen kokoamiseen (bruttopalkka, verot, vähennykset, nettopalkka).
* Palkkalaskelman tietojen tallentaminen local storageen (esim. myPaychecks).
* Käyttöliittymäkomponentin luominen palkkalaskelman näyttämiseen.

**Viikko 3: Budjetointi**

**Budjettitietojen Malli ja Tallennus (Local Storage):**

* Funktioiden luominen budjettitietojen (kategoria, ajanjakso, summa) tallentamiseen, hakemiseen, päivittämiseen ja poistamiseen local storagessa (esim. myBudgets).

**Budjetin Luomisen Käyttöliittymä:**

* Lomakkeen luominen budjetin luomiseen ja muokkaamiseen.
* Tietojen synkronointi local storagen kanssa.

**Tulo- ja Menotapahtumien Malli ja Tallennus (Local Storage):**

* Funktioiden luominen tulo- ja menotapahtumien (kategoria, summa, päivämäärä) tallentamiseen, hakemiseen, päivittämiseen ja poistamiseen local storagessa (esim. myTransactions).

**Tulojen ja Menojen Syöttö ja Listaus:**

* Käyttöliittymäkomponenttien luominen tulo- ja menotapahtumien syöttämiseen ja listaukseen.
* Tietojen synkronointi local storagen kanssa.

**Budjetin ja Toteutuneiden Lukujen Vertailu:**

* Logiikan luominen budjetin ja toteutuneiden tulojen/menojen vertailuun (tiedot haetaan local storagesta).

**Raportointi:**

* Yleiskäyttöisten raportointikomponenttien luominen (esim. taulukot).
* Kaavion luominen (joko itse tai käyttäen kirjastoa, esim. Chart.js).
* Palkkalaskelmien listaus
* Budjettiyhteenveto.
* Budjetin ja toteutuman vertailu (esim. taulukko, kaavio).
* Menojakauma (esim. ympyrädiagrammi).

**Viikko 4: Testaus, Viimeistely ja Julkaisu**

**Testaus:**

* Yksikkötestien kirjoittaminen kriittisille logiikkakomponenteille (palkanlaskenta, budjetin vertailu).
* Integraatiotestien kirjoittaminen (käyttöliittymä <-> local storage).
* Käyttöliittymätestaus (eri selaimilla ja laitteilla).

**Virheenkorjaus ja Optimointi:**

* Havaittujen virheiden korjaaminen.
* Koodin siistiminen ja kommentointi.
* Suorituskyvyn optimointi (tarvittaessa).

**Dokumentaatio:**

* Käyttöohjeiden kirjoittaminen.
* Yksinkertaistetun teknisen dokumentaation laatiminen (esim. README.md-tiedoston täydentäminen).
* Local storagen rajoitusten mainitseminen dokumentaatiossa.