

Opintojakson nimi

Koulutus

Ryhmä

Palautuspäivämäärä

Jouni Kiviperä

# Älykäs Aikataulutusassistentti

Tehtävienhallinta ja Google Kalenteri -integraatio



# **Sisältö**

<b>Käytetyt lyhenteet tai sanasto</b>	<b>3</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 Key Features and Functionality</b>	<b>5</b>
<b>3 Use Cases</b>	<b>6</b>
<b>4 Ethical Considerations</b>	<b>7</b>
<b>5 Risks and Challenges</b>	<b>8</b>
<b>6 Benefits</b>	<b>9</b>
<b>7 Technical Details</b>	<b>10</b>
<b>8 Demonstration (if applicable)</b>	<b>11</b>
<b>9 Future Trends</b>	<b>12</b>
<b>10 Conclusion</b>	<b>13</b>
<b>11 References</b>	<b>14</b>

## Käytetyt lyhenteet tai sanasto

Lyhenne	Lyhenteen selitys
API	Application Programming Interface – ohjelmointirajapinta
AI	Artificial Intelligence – tekoäly
JSON	JavaScript Object Notation – kevyt tietomuoto
CLI	Command Line Interface – komentorivikäyttöliittymä
Google API	Googlen tarjoama ohjelmointirajapinta eri palveluille
OAuth 2.0	Avoim autentikointistandardi, jota käytetään Google API:n kirjautumisessa
Scikit-learn	Python-kirjasto koneoppimiseen
ML	Machine Learning – koneoppiminen
PuLP	Python-kirjasto lineaariseen optimointiin
Python	Korkean tason ohjelmointikieli, jota käytetään projektissa
Tekoäly	Algoritmit, jotka voivat oppia ja tehdä päätöksiä
Task Manager	Sovelluksen moduuli, joka hallinnoi tehtäviä
Google Calendar Sync	Sovelluksen moduuli, joka synkronoi tehtävät Google Kalenteriin

# 1 Introduction

Älykäs aikataulutusassistentti on Pythonilla toteutettu sovellus, joka käyttää tekoälyä ja Google Calendar API:a tehostamaan käyttäjän ajanhallintaa. Sovellus optimoi tehtävien priorisoinnin, ajoituksen ja synkronoinnin Google Kalenteriin.

Moderni elämä on kiireistä, ja tehtävien hallinta voi olla haasteellista. Tämä sovellus ratkaisee ongelman tarjoamalla älykkään, käyttäjälle mukautuvan ratkaisun, joka automatisoi ja optimoi aikataulutuksen.

## 2 Key Features and Functionality

Älykäs aikataulutusassistentti on Pythonilla toteutettu sovellus, joka käyttää tekoälyä ja Google Calendar API:a tehostamaan käyttäjän ajanhallintaa.

Sovellus optimoi tehtävien priorisoinnin, ajoituksen ja synkronoinnin Google Kalenteriin.

Moderni elämä on kiireistä, ja tehtävien hallinta voi olla haasteellista. Tämä sovellus ratkaisee ongelman tarjoamalla älykkään, käyttäjälle mukautuvan ratkaisun, joka automatisoi ja optimoi aikataulutuksen.

### 3 Use Cases

Opiskelija: Hallitsee kurssitehtäviä, tenttejä ja projekteja.

Työntekijä: Suunnittelee työtehtäviä ja tapaamisia kiireisessä ympäristössä.

Sovellus parantaa tuottavuutta ja auttaa käyttäjiä keskittymään tärkeimpiin tehtäviin.

## 4 Ethical Considerations

Sovellus käyttää avoimia algoritmeja, eikä se tee päätöksiä käyttäjän henkilökohtaisten ominaisuuksien perusteella.

Käyttäjätiedot, kuten kalenteritiedot, käsitellään turvallisesti ja noudattaen Google API:n tietoturvakäytäntöjä.

## 5 Risks and Challenges

Sovellus käyttää avoimia algoritmeja, eikä se tee päätöksiä käyttäjän henkilökohtaisten ominaisuuksien perusteella.

Käyttäjätiedot, kuten kalenteritiedot, käsitellään turvallisesti ja noudattaen Google API:n tietoturvakäytäntöjä.



## 6 Benefits

Käyttjäedut: Automatisoi tehtävien hallinnan ja vähentää stressiä. Tehostaa ajankäyttöä priorisoimalla tärkeät tehtävät.

Vaikutus teollisuuteen: Sovellus tarjoaa skaalautuvan ratkaisun henkilökohtaiseen ja työelämän ajanhallintaan.

## 7 Technical Details

Algoritmit: Koneoppiminen käyttäjän tapojen oppimiseen (Scikit-learn).

Lineaarinen optimointi tehtävien sijoittamiseen (PuLP).

Datan käsittely: Tehtävät tallennetaan paikallisesti JSON-muodossa. Google Calendar API:n kautta tapahtumat synkronoidaan

## **8 Demonstration (if applicable)**

Demo näyttää, miten käyttäjä syöttää tehtäviä ja miten ne optimoidaan ja synkronoidaan kalenteriin. Tulokset visualisoidaan kaavioilla.

## 9 Future Trends

Tulevat ominaisuudet: Graafinen käyttöliittymä (Tkinter tai Flask). Syvempi koneoppiminen käyttäjän toiminnan analysointiin.

Teollisuustrendit: Tekoälyn käyttö henkilökohtaisessa ajanhallinnassa kasvaa. API-pohjaiset ratkaisut ovat yhä tärkeämpiä.

## 10 Conclusion

Tulevat ominaisuudet:\*\* Graafinen käyttöliittymä (Tkinter tai Flask). Syvempi koneoppiminen käyttäjän toiminnan analysointiin.

Teollisuustrendit:\*\* Tekoälyn käyttö henkilökohtaisessa ajanhallinnassa kasvaa. API-pohjaiset ratkaisut ovat yhä tärkeämpiä.

## 11 References



