

PC HARDWARE SPECIFICATIONS FOR A REQUIRED SOFTWARE

Federico Dutto – Thomas Pashollari 5°A Inf

Obiettivo del progetto:

Realizzare una web app che permetta di ottenere i requisiti minimi hardware per un determinato software preso in input.

Materiale:

1. Linguaggi utilizzati

Attualmente, date le nostre competenze in ambito informatico, per realizzare il nostro progetto il linguaggio che consiglio di utilizzare è **PHP**. Questo linguaggio ci servirà per gestire il backend dell'applicazione, grazie soprattutto alle sue caratteristiche che lo hanno contraddistinto negli anni.

Infatti PHP è noto per la semplicità d'uso, per avere un ampio supporto per i database, un numerosissimo numero di estensioni/librerie e una buona gestione della sicurezza e delle sessioni. Inoltre è stato scelto questo linguaggio in quanto garantisce un'ottima velocità di esecuzione.

2. Framework di sviluppo

Lavorando con PHP, il framework che utilizzeremo sarà **Laravel**, in quanto possiede le caratteristiche adatte al progetto che stiamo realizzando.

Per prima cosa, troviamo Eloquent, un sistema di ORM che serve a semplificare l'interfacciamento con i nostri DB. Inoltre questo framework implementa una gestione del middleware che permette di filtrare le richieste HTTP. Questo inoltre ci aiuta con il buon sistema di autenticazione e di gestione delle sessioni che Laravel offre.

3. Librerie con cui lavorare & API

WEB scraping API, possibilmente supportate da AI, per logica applicativa, in modo da ottenere specifiche dal web. Sono utilizzate per estrarre da pagine HTML determinate informazioni desiderate. Questa operazione, fondamentale per raggiungere l'obiettivo del progetto, è denominata “**parsing HTML**”.

4. Strumenti di autenticazione

Da realizzare eventualmente in aggiunta, una volta concluso l'obiettivo principale.

Un buon strumento per la gestione dell'autenticazione è la libreria **OAuth** (Open Authorization). Questo sistema si appoggia su un servizio di terze parti (Authorization Server) per gestire il processo di autorizzazione e rilascio dei token di accesso. Inoltre supporta l'utilizzo del protocollo HTTPS.

5. DB relazionale

Utilizzeremo **MariaDB**, che è un sistema di gestione di **database relazionali** open source (è una fork di MySQL).

La scelta è stata dettata dalla buona integrazione con il framework Laravel, grazie anche allo strato software Eloquent. Inoltre include funzionalità di **sicurezza avanzate**, come la crittografia dei dati, l'accesso basato su ruoli e altre caratteristiche per proteggere il database da accessi non autorizzati.

6. Strumenti di deploy

Per deployare la nostra applicazione, utilizzeremo **AWS** come piattaforma di hosting. Questa piattaforma fornisce servizi scalabili in modo da aumentare o diminuire le risorse in base al carico, permettendo il pagamento solo di quello che si consuma (Pay as you go).

I principali vantaggi di questo servizio di Amazon riguardano l'infrastruttura distribuita a livello globale, che consente di posizionare le risorse vicino agli utenti per migliorare le performance, e l'adattamento ai picchi di carico.

7. Versioning, unit test & continuous integration

- **Git & GitHub:** classico sistema di versioning con il suo salvataggio remoto
- **PHPUnit:** fornisce strumenti per eseguire gli unit test nel linguaggio php
- **Jenkins:** permette le pratiche di continuous integration (CI) e continuous deployment (CD), ed è inoltre possibile integrarlo con GitHub per automatizzare questi processi di sviluppo.