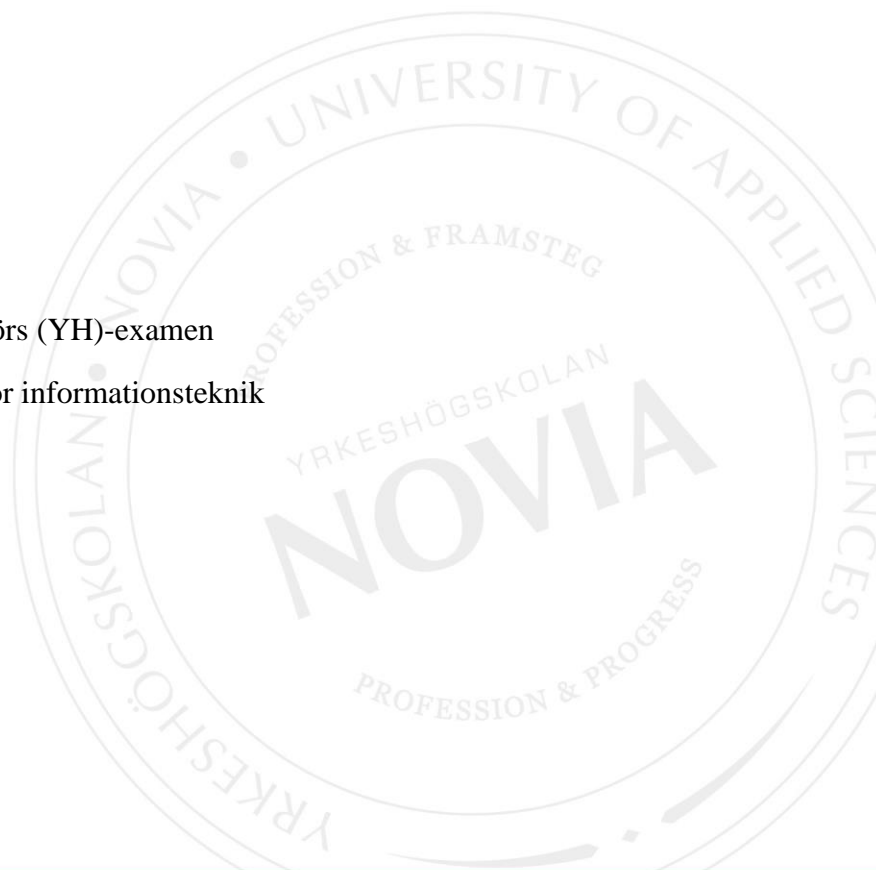


# Utveckling av digital produktivetsregistrering vid Närpes Grönsaker

Samir Vehabovic

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för informationsteknik  
Vasa 2021



## EXAMENSARBETE

Författare: Samir Vehabovic  
Utbildning och ort: Informationsteknik, Vasa  
Handledare: Kaj Wikman

Titel: Utveckling av digital produktivetsregistrering vid Närpes Grönsaker

---

Datum: 28.3.2021

Sidantal: 21

---

### Abstrakt

Detta examensarbete gjordes för Närpes Grönsaker som är ett packeri av inhemska växthusgrönsaker såsom gurka, tomater, paprika, chili och specialtomater. Examensarbetet handlar om insamling och hantering av data via ett webbgränssnitt och lagring till en databas. Syftet var att ha en mer digitaliserad webbapplikation som skall vara lätt att använd.

Detta åstadkoms genom att skapa en webbapplikation för att både förbättra det föregående systemet och göra informationen bättre tillgänglig. Projekt gjordes med Apache, MYSQL, PHP och JavaScript.

Resultatet är en färdig webbsida som skall kunna användas ute i produktionen och administrationen skall enkelt kunna exportera informationen till Excel-dokument.

---

Språk: svenska

Nyckelord: webbapplikation, PHP, JavaScript

---

## BACHELOR'S THESIS

Author: Samir Vehabovic  
Degree Programme: Information Technology, Vaasa  
Supervisor: Kaj Wikman

Title: Development of Digital Productivity Registration at Närpes Grönsaker

---

Date: March 28, 2021

Number of pages 21

---

### **Abstract**

This thesis was done for Närpes Grönsaker which is a packing company of homegrown greenhouse vegetables such as cucumber, tomatoes, paprika and chili. The thesis is about collecting and managing data that is collected through a web interface and stored to a database. The purpose was to have a more digitized web application that is easy to use.

This was accomplished through creating a web application to improve its predecessor and to make the information better available. The project was created with the use of Apache, MYSQL, PHP and JavaScript.

The result is a finished website that can be used in production to view and administer the data. The administrator can easily export the information to an Excel document.

---

Language: Swedish

Key words: Web Application, PHP, JavaScript

---

# Innehållsförteckning

1	Inledning.....	1
1.1	Beskrivning på uppgiften .....	1
1.2	Bakgrund .....	2
1.3	Mål.....	2
1.4	Uppdragsgivare.....	2
1.5	Maskinpresentation.....	4
2	Teori .....	5
2.1	MySQL.....	5
2.2	PHP.....	5
2.3	HTML.....	6
2.4	CSS.....	7
2.5	JavaScript (JS) .....	7
2.6	XML .....	8
2.7	Virtuell maskin .....	8
2.8	Ubuntu .....	9
3	Metoder .....	10
3.1	Skapa från grunden .....	10
3.2	Ramverk.....	10
3.3	CMS .....	10
4	Lösning.....	11
4.1	Analys.....	11
4.2	Prototyp och utvecklingsmiljö .....	12
4.3	Virtuell webbserver.....	13
4.4	Databasen.....	13
4.5	Import av filer.....	14
4.6	Webbapplikationen .....	15
4.7	Innehåll i webbsidorna .....	15
5	Resultat .....	17
6	Diskussion.....	20
7	Referenser.....	21

# 1 Inledning

Detta examensarbete avser ett projekt som har beställts av Andelslaget Närpes Grönsaker som är ett lokalt företag i Österbotten. Tjänsten som behövdes var en webbapplikation där man skall kunna fylla i uppgifter som gäller produktionen och spara detta till en databas.

Uppgifterna baserade sig på olika partier som packas under dagens lopp för varje enskild produktionslinje. Man skall också kunna exportera dessa uppgifter till Excel för att göra kunna göra vidare uppföljningar.

För att uppnå detta behövdes en webbapplikation som samlar in, sparar och presenterar information. Detta gjordes i flera steg där man planerade, byggde, testade och implementerade.

Detta system skulle vara enkelt att lära ut och använda för slutanvändaren.

## 1.1 Beskrivning på uppgiften

Mitt examensarbete är ett projekt som skall samla in information spara den till en databas och göra det möjligt för en användare att redigera eller lägga till mera information. Detta betyder att man måste ha ett system för insamling av data via ett webbgränssnitt och ett för att spara all information till en databas. Projektet krävde att det skapades en webbserver i detta fall valdes Ubuntu virtuell dator där man har webbserver, databasen och alla filer som ingick i projektet efter att det var avklarat.

Eftersom de datorer som hanterar maskinerna man använder i produktionen kan spara all möjlig data den samlar in under dagen som man packar produkterna. Datorerna har program som kan skapa XML-filer som man har använt för att spara till andra databaser i kapitel 2.6 förklaras det mer om vad XML är. Man kommer att behöva kunna använda dessa XML-filer och ha ett automatiserat system för överföring till den databasen som detta projekt kommer att använda och som skall kunna behandla informationen och spara till databasen.

## 1.2 Bakgrund

I nuläget har man haft ett system där man fyller vart efter man packar till en Excel tabell, som har ett färdigt upplägg och detta används då man sparar nya data. Detta sätt har haft sina brister och fördelar men man ville byta till ett webbgränssnitt och databas, för att kunna vidareutveckla detta och utnyttja data bättre för uppföljningar med mera.

Ett av de större problemen med Excel-dokumentet har varit att man inte har kunnat spara mer detaljerad beskrivning av de problem som uppstår under dagen med maskinen och andra saker som gör att produktionen kan rulla på. Istället behövde man ett system där man kan göra mer detaljerade beskrivningar för varje parti som man har packat under dagen.

Efter att ha gjort en analys av nuläget skapades en prototyp för testning.

## 1.3 Mål

Målet med arbetet var att analysera, planera och förverkliga ett webbaserat system.

- Göra en analys för att se vad som exakt finns nu och vad som behöver ändras.
- Systemet skall automatiskt implementera data från XML-filer.
- Olika kontroller i webbgränssnittet att allt fylls i rätt och att inget glöms.

Dokumentation:

- Skall finnas dokumentation över det man har skapat för framtida ändringar.
- En användarguide på hur man skall använda det enkelt och effektivt.

## 1.4 Uppdragsgivare

Som uppdragsgivare för arbetet var Andelslaget Närpes Grönsaker, företaget är beläget i Närpes. Företaget grundades 1957 av 17 odlare närvarande, man hade mötet i Böle där man höll ett konstituerande möte och bildandet av Närpes Grönsaker. Orsaken man gick in för att bilda företaget var missnöjet med den dåvarande situationen och en lokal partiaffären som många av odlarna sålde till upphörde med sin verksamhet.

De första två åren var ganska svåra för företaget, man hade några större kreditförluster sedan hade man också svårt med den lokala konkurrensen. Tack var finansiären Robert Mattlar och den dåvarande nyanställda Konrad Perus som VD så lyckades man vända utvecklingen. Till en början packade odlarna själva sina produkter men efter 10 års utveckling byggde man sitt första egna utrymme.

Efter sitt andra årtionde hade man bytt ut VD eftersom den gamla gick i pension, man byggde en hall på tomt som man hade införskaffat.

Med sitt tredje årtionde fortsatte man utvecklas man utvidgade företaget med större hall på 1600 kvadratmeter, trähäckar byttes ut mot plast häckar för råvarorna och det införskaffades nya maskiner för gurkförpackning, tomatförpackning och lådresare för tomlådor (Sörhannus, 2007).

Man har genom åren utvecklats med modernare maskiner, utvidgad hallen för att hantera större produktionskrav och stärkt sin position på marknaden. Kvaliteten är fortsatt viktig och 2018 utvecklades verksamheten med ISO 22000 certifikat.



Figur 1. Bild över företaget Närpes Grönsaker.

## 1.5 Maskinpresentation

De maskiner som används vid Närpes Grönsaker för paprika, tomat och gurka är ganska liknande eftersom de har gjorts av samma tillverkare, Greefa från Holland. Av dessa maskiner är tomat och gurka maskinen de största i företaget. Paprika har en enklare version av maskinen, där man sorterar vad som är första klass, andra klass och avfall. Första klass paprika kommer ut från maskinen enligt specifika storlekar till olika utgångar.

Tomat maskinen har inmatning där man kör in en hel pall full av häckar som är staplade till viss höjd, maskinen staplar ner häckarna till en och en. Dessa häckar blir tömda och körs vidare via ett sorteringsband, där två eller flera personer sorterar, andra klass och avfall. Maskinen lägger in tomaterna till lådor och kör ut dessa lådor till två olika robotar som staplar lådorna på en pall till en bestämd höjd. Slutligen så plastas pallen in och körs ut från roboten där en person hämtar den färdiga pallen och kör till utlastningshallen.

På gurka ser denna process likande ut men där har man bara en robot som staplar upp de färdiga lådorna på pallen, och man har utgångar där några personer måste lägga gurkorna i lådor manuellt. Tomat och gurka har flera datorer som gör styrningar och samlar upp information och allt detta sparas på en server. De program som Greefa har skapat kan använda informationen som maskinen samlar upp för att skapa till exempel XML-dokument för varje parti, som körs igenom maskinen, eller för att skapa en html baserad rapport, vad html är förklaras i kapitel 2.3.

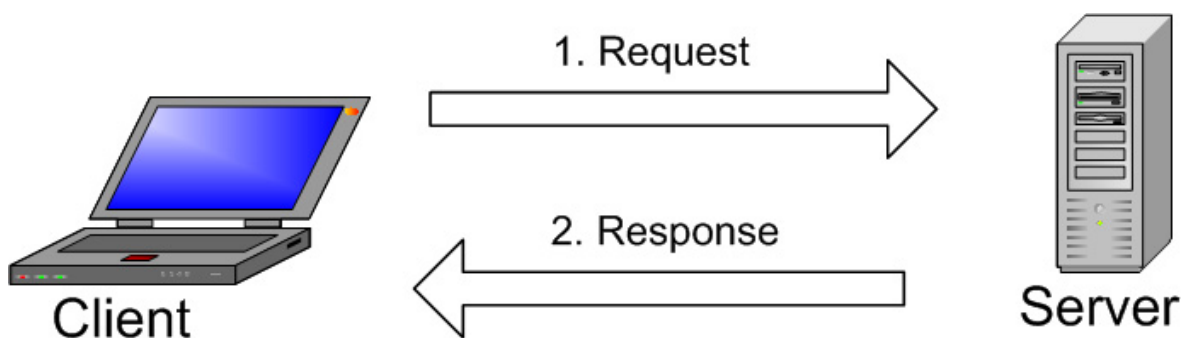


## 2 Teori

Detta kapitel är en introduktion till de verktyg, språk och grunder som användes för detta examensarbete.

### 2.1 MySQL

MySQL är öppen källkod som tillåter behandling av relationsdatabaser och är baserat på SQL (Structured Query Language). MySQL skapades av ett svenskt företag vid namnet MySQL AB, 1994, MySQL kan köras på nästan vilken plattform som helst och har blivit väldigt populärt för att det är lätt att använda, säkert och är industristandard eftersom det har använts så länge. MySQL baserar sig på klient-servermodell. Grunden för MySQL är MySQL -server som tar hand om alla databasinstruktioner. MySQL arbetar med olika tillägsprogram som tar hand om administrativa delen av MySQL (Herawan, 2020).



Figur 2. Bild från hostinger.com.

### 2.2 PHP

PHP akronym för Hypertext Preprocessor är ett öppen källkod skript språk som är lämpat för webbutveckling och kan integreras i html koden. Det är byggt som ett verktyg för server-sidig kommunikation, det menas att den utför koden på serversidan som sedan skickas till klienten. Detta gör att klienten får resultatet men vet inte vilken kod man använde för att uppnå detta. PHP skapades 1995 av Rasmus Lerdorf för personlig användning och har utvecklats för möjlighet av objektorienterad programmering (Bilqis, 2020). Ett exempel på en PHP kod finns i bilen nedan, koden skriver ut ”Hi, I’m a PHP-script”.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

Figur 3. Kodexempel taken från php.net

## 2.3 HTML

HTML står för Hyper Text Markup Language som skapades 1991 av Tim Berners-Lee och eftersom det blev rätt snabbt populärt så har det blivit officiell webstandard. HTML används främst inom skapelse av webbsidor och det har varit många versioner under åren, allt eftersom man har uppdaterat så har ny version utvecklats, nyaste är HTML5 (Oxford Web Studio, 2020).

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Example page</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is an example of a basic HTML page.</p>
  </body>
</html>
```

Figur 4. Kod exemplar tagen från w3schools.

HTML är ett metaspråk, inte ett programmeringsspråk, eftersom det används taggar som vanligtvis kommer parvis en som öppnar och en som stänger, och har inte möjligheten att skapa dynamiska funktioner. HTML tillåter skapelse av strukturerade sektioner, paragrafer, länkar och rubriker för webbsidor och applikationer.

## 2.4 CSS

CSS betyder Cascading Style Sheet och är språk för att beskriva presentations stilen. Det separerar den visuella stilen från innehållet. CSS skapades 1996 av W3C (World Wide Web Consortium) för att sidorna började bli komplexa och HTML som gjorde detta från början började skapa problem för utvecklare. Eftersom HTML var grunden för hemsidan och CSS som tar hand om stilen gör att detta två går hand i hand (Hostinger, 2020).

CSS kan ta hand om storleken, färgerna, skrivstil, var objektet skall vara med mera. CSS kan användas för att forma allt individuellt och man kan använda samma template för att forma flera sidor. Om man behöver ändra något så är det bara att ändra i CSS filen och uppdatera sidan som ändringen gäller för utan desto mer krångel. Att kunna CSS är tekniskt inte viktig del av webbprogrammering men för att kunna skapa en sida med stil behövs detta. Exemplet nedanför är ett sätt att forma rubriks taggen H1.

```
h1 {  
color: blue;  
text-align: left;  
}
```

## 2.5 JavaScript (JS)

JS är ett av de populärare programmeringsspråken (Hostinger, 2020) och en av de tre huvudspråken som används vid webbprogrammering. JS skapades 1995 av Brendan Eich, dåvarande Netscape anställd och var i början exklusiv för Netscape. Språket hette i början livescript, men tack vare att språket växte hela tiden skapades nutida JavaScript som fungerar både på olika webbläsare men också på olika enheter, och är ett av tre viktiga språk för webbutveckling. Det finns även många JavaScript bibliotek som har skapats för att underlätta. Med JS kan man skapa allt möjligt från dynamiskt uppdaterande innehåll, kontrollera multimedia, animera bilder och mycket mer.

```

1  function greetMe(yourName) {
2      alert('Hello ' + yourName);
3  }
4
5  greetMe('World');

```

Figur 5. Bild från mozilla.org.

## 2.6 XML

XML står för eXtensible Markup Language utvecklades för att datorer och människor skall enkelt kunna läsa av genom att använda olika taggar och varje av dem har logisk betydelse. Taggar som skapas är unika och skapas av användaren (Oxford Web Studio, 2020). XML är ett metaspråk ett språk som tillåter att skapa och definiera andra språk.

```

<?xml version="1.0" encoding="UTF-8"?>
<GREEFA>
  <ProduktKod>2201</ProduktKod>
  <AllmäntNummer>20190429-1-001</AllmäntNummer>
  <Leveransdatum>26-04-2019</Leveransdatum>
  <Sorteringsdatum>29-04-2019</Sorteringsdatum>
  <Odlarnummer>162 </Odlarnummer>
  <Odlarnamn> </Odlarnamn>
  <Antalhäckar>53</Antalhäckar>
  <TotaltVikt>747 </TotaltVikt>
  <IKlassKg>730 </IKlassKg>
  <IIKlassKg>15 </IIKlassKg>
  <AvfallKg>3 </AvfallKg>
  <medelfruktvikt>368</medelfruktvikt>
  <medellängd/>
  <NummerIKlass>1982 </NummerIKlass>
  <Operatör>Samuel</Operatör>
  <Noteringar/>
  <Anmärkningar/>
</GREEFA>

```

Figur 6. Bild av XML-fil från Närpes Grönsaker.

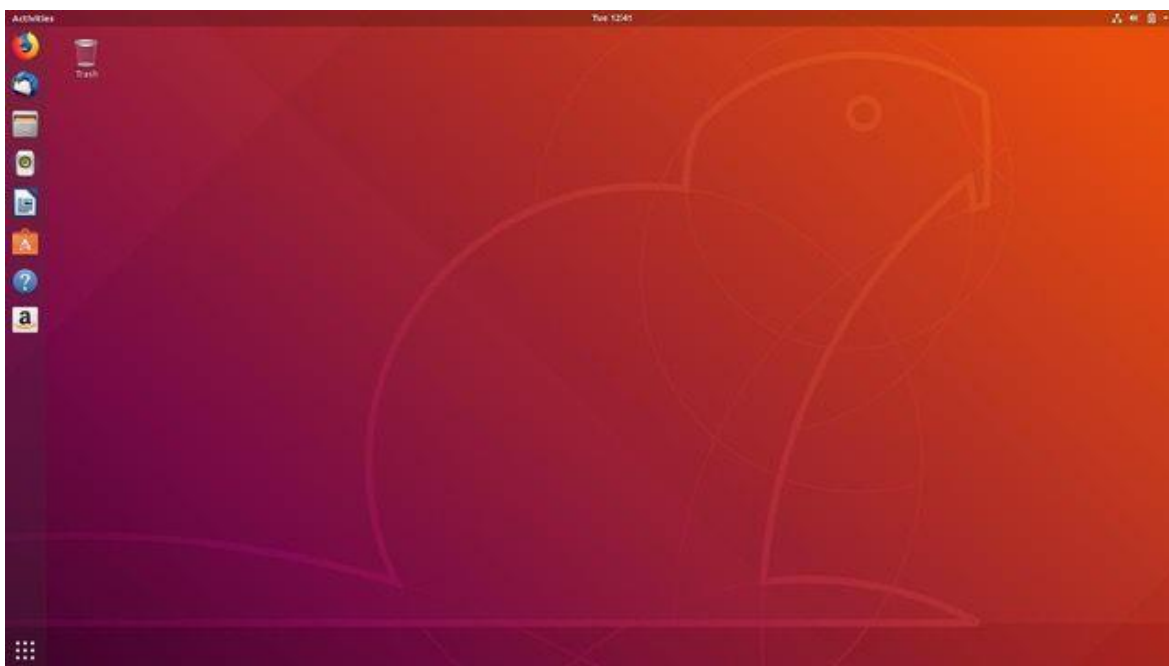
## 2.7 Virtuellt maskin

En virtuell maskin är en virtuell miljö som har skapats genom programvara. Virtuella miljön körs i ett separat fönster som andra program och användaren får samma upplevelse på den virtuella maskinen som på värdoperativsystemet (Microsoft, 2020). På datorn används vanligen ett operativsystem och man kan ha flera virtuella maskiner med olika operativsystem. Den virtuella miljön brukar ha fördelar att vad man än gör där så speglas det

inte på värddatorn, detta brukar man utnyttja för att testa betaprogram med mera. Då man skapar ny virtuell dator så får den dela på de resurser som värddatorn har. För att kunna köra en virtuell maskin måste virtualisering vara igång på moderkortet.

## 2.8 Ubuntu

Ubuntu är ett operativsystem som är baserat på Linux kärna. Systemet är helt gratis för både privat användning som för företag. Deras motto har alltid varit att de kommer vara gratis för alltid och de skapar nya versioner hela tiden där man vartannat år skapar en version som har längre tids support (Ubuntu, 2020). Linux är ett system som har utvecklats av Linus Torvalds och är baserat på ett Unix operativsystemet. Linux är öppen källkod där alla kan se källkoden och lämna bidrag till framtida versioner (Debian, 2020).



Figur 7. Bild på Ubuntu Systemet.

## 3 Metoder

Allt eftersom saker och ting blir mer digitala så har det blivit ett måste att ha en egen hemsida både en för utomstående som kan besöka men även interna sidor för att ha information sparad och uppvisad på. Många väljer att bygga sidor som är anpassade både till mobila enheter och datorer. Det finns flera olika ramverk, CMS (Content Management System) och verktyg att skapa en hemsida med. Här tas upp ett par av dem.

### 3.1 Skapa från grunden

Man kan skapa en hemsida helt från grunden själv, med andra ord koda allt själv. Man kan använda template där man får grunden för utseendet, men man måste programmera alla funktioner som man vill ha själv. För att skapa en hemsida från grunden så kan man använda HTML, PHP, CSS och olika bibliotek så som Bootstrap för utseendet som man kan också hämta färdiga templates ifrån. För det dynamiska innehållet och funktioner så kan man använda JavaScript, Ajax eller jQuery. Det finns olika fördelar och nackdelar att skapa webbsidan själv men man väljer vad som passar till den slutgiltiga produkten.

### 3.2 Ramverk

Ramverk som kodare använder till för att bygga webbsidor, är ett tillägg till ett programmeringsspråk och som innehåller en uppsättning bibliotek (Berezhnoi, 2020). Ramverk kräver att man har goda programmerings kunskaper men ger också stora möjligheter att skapa skräddarsydda webbsidor och webbapplikationer. Asp.net är ett av dessa ramverk som bygger på programmeringsspråket C# men som också hanterar PHP och bibliotek som JavaScript med mera. Asp.net är skapad av Microsoft och är skapad från början till Windows men det går att köra på andra system också

### 3.3 CMS

CMS-system som står för innehållshanteringssystem är egentligen en programvara som hjälper till att bygga webbsidor och webbapplikationer. Där man får en nästan färdig

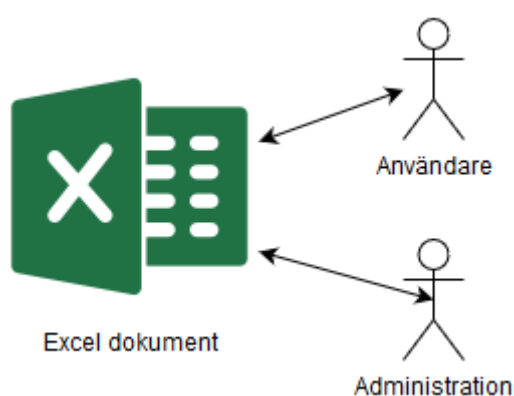
hemsida men som behöver ändras enligt egna behov och det finns färdig kodade tillägg för alla möjliga saker. Man behöver bara ladda upp applikationen till webbservern, det finns en första installation och efter det är det bara att ändra utseendet och använda eller koda själv tillägg som behövs. Wordpress är ett av de öppna CMS-systemen som används ganska mycket idag, enligt dem själva så har 35 % av webben Wordpress som bas för sin hemsida (Wordpress, 2020). Eftersom så pass många använder Wordpress finns det många olika teman att välja mellan och tillägg för att skapa webbsidan. De har också en administrations panel där man styr själva sidan ifrån just den del som besökare ser då de kommer in på sidan.

## 4 Lösning

I följande kapitel kommer det att tas upp vilka steg som gjordes för att kunna skapa webbapplikationen.

### 4.1 Analys

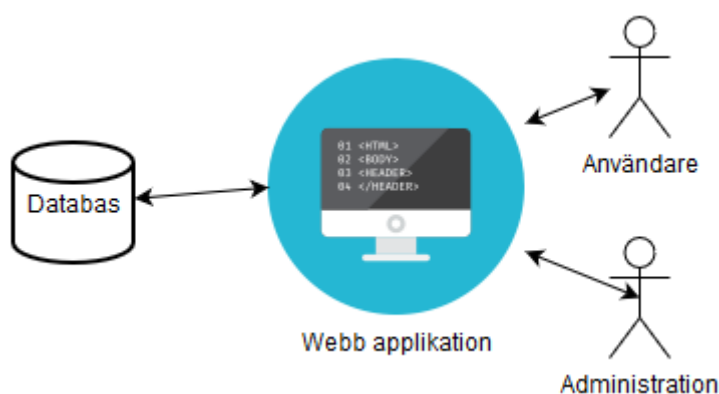
För att få fram en klarare bild över nuvarande läget och hur det har gått till gjordes en analys. I företaget har man från förr införskaffat en intern websida för lagring av partier på det så kallade special packet där biff, körsbär, Dunne, pärltomater med mera hör till. Och där skapar man nytt inlägg för varje parti man packar. Man ville ha liknande system på de resterande maskinerna där man gjorde samma rapport men bara i Excel.



Figur 8. Beskrivning av användning av Excel-dokument.

Eftersom rapporten var gjord i Excel så blev dokumentet låst då användaren på linjen hade dokumentet öppet hela tiden. Ofta glömde man att stänga dokumentet efter skiftets slut och

att spara ibland. En av orsaken man ville ha det webbaserat var att sådana misstag skulle undvikas och att man skulle ha allt sparad till en databas. Men skillnaden var att maskinerna man har införskaffat från maskintillverkaren Greefa för paprika, tomat och gurka har möjlighet att samla in information över de produkter som maskinen packar, men också skapa XML-filer via deras egen programvara. Dessa innehåller information på olika saker och man ville utnyttja detta också. I den nuvarande formen där Excel använder har man själv fått mata in totala kilo förstaklass, andraklass och avfall, men detta finns redan i XML-filerna, man matade också in information för en hel dag med alla störningar i produktionen och pauser.



**Figur 9. Beskrivning på hur det kommer att fungera i praktiken.**

Eftersom Greefa-maskinen skapar XML-filer behövdes också ett automatiserat system som importerar denna data direkt till databasen och sedan skall användaren kunna fylla på med information över det som maskinen inte täcker.

## 4.2 Prototyp och utvecklingsmiljö

Utvecklingen av webbapplikationen gjordes på en hemma dator som hade alla de verktyg som behövdes före man laddade upp allt till servern. Programmet WAMPserver användes som är ett gratis program och möjliggör utveckling av webbapplikationer, eftersom den innehåller Apache, PHP och MySQL i nästan färdig konfigurerad miljö och installeras enkelt på Windows.

Kodningen gjordes med hjälp av Visual Studio Code (Vscode), det finns olika program att använda men i detta fall valdes detta program. Vscode har pakethanteringssystem med många tillägg som har skapats av gemenskapen, med dessa kan man använda för att hitta fel i koden som man eventuellt gör och har också möjlighet att snygga till koden för att man skall ha det enklare att lägga till saker. Vscode är ett gratis program under MIT licensen och



är skapat av Microsoft, den är lätt hanterlig av operativsystemet och har stöd för flera olika operativsystem.

Prototypen skapades steg för steg.

1. Skapelse av databasen och fastställa namnen på innehållet i tabellerna och eventuella relationer mellan tabellerna.
2. Skapelse av import skriptet mellan två datorer och av PHP-skriptet som skulle importera allt till databasen.
3. Skapelse av webbsidorna som skall användas och uppföljning för att se om allt skapas enligt kundens önskemål.
4. Live testning och buggkorrigeringar.

Efter slutförd prototyp gjordes en version som skall vara i användning.

### **4.3 Virtuellt webbserver**

För att undvika att data förloras eller att man tar bort filer som används av andra system så skapades en virtuellt Ubuntu 18.04 maskin som server och lagring för allt. En webbserver som innehåller Apache, HTTP Server, MySQL, PHP och samba. Samba installerades eftersom man behöver dela vissa mappar där XML-filerna skall kopieras till. Då det är egen server gör detta att de blir inga störningar på resten av serverna, och man har rättighet att lättare kunna installera och skapa allt som behövs för att slutanvändaren skall kunna använda systemet. Installation av helt ny Ubuntu server skedde via VirtualBox och installerade in alla de program som behövdes. På detta sätt uppnådde man också egna mappar för lagring av XML, och man kan ta bort de filer vart eftersom de importeras till databasen.

### **4.4 Databasen**

Databasen skapades i MySQL och tabellerna designades att de skulle kunna ta emot alla fält per produktionslinje som man behövde data för. Som primär nyckel för gurka, tomat och paprika tabellen användes ett unikt nummerformat som importeras från XML-filerna. Medan för flowpack användes samma nummer men skapades dynamiskt av webbapplikationen. Detta nummer består av dagens datum, mellanstreck och tre numror som börjar från nummer

ett varje ny dag. Till en början skapades flera tabeller för varje enskild linje där man skulle ha en tabell för huvudinmatningen, en för alla driftstoppval och den sista som sparar driftstopp valet och minuterna. Men kunden ville att varje linje skulle ha en enda tabell som behandlar detta, vilket gör att databasen kan bli ganska stor i framtiden eftersom vissa värden måste sparas även om de inte används.

Tabellen för tomat ser ut följande på Figur 9 där man har använt samma grund för alla de andra tabellerna som behövdes med vissa ändringar och vissa saker som de andra linjerna har annorlunda.

AllmantNummer : varchar(20)	
AllmantSort : varchar(50)	DS4 : varchar(25)
Odlarnummer : int(11)	DSm4 : int(11)
Odlarnamn : varchar(30)	DSk4 : int(11)
ProduktKod : int(11)	DS5 : varchar(25)
Leveransdatum : date	DSm5 : int(11)
Antalhackar : int(11)	DSk5 : int(11)
Partifordelning : varchar(20)	DS6 : varchar(25)
LinjeForare : varchar(20)	DSm6 : int(11)
PersonerLinje : int(11)	DSk6 : int(11)
Sorteringsdatum : date	DS7 : varchar(25)
SorteringStartTid : time	DSm7 : int(11)
SorteringStopTid : time	DSk7 : int(11)
KaffePaus : int(11)	DS8 : varchar(25)
Matpaus : int(11)	DSm8 : int(11)
AnnanPausMin : int(11)	DSk8 : varchar(50)
OrsakAnnan : varchar(255)	IKlassKg : int(11)
DS1 : varchar(25)	IIKlassKg : int(11)
DSm1 : int(11)	AvfallKg : int(11)
DSk1 : int(11)	TotalVikt : int(11)
DS2 : varchar(25)	medelfruktvikt : int(11)
DSm2 : int(11)	RodaTomater : decimal(5,2)
DSk2 : int(11)	GronaTomater : decimal(5,2)
DS3 : varchar(25)	Anmarkningar : varchar(50)
DSm3 : int(11)	Status : int(11)
DSk3 : int(11)	Hastighet : int(11)

Figur 10. Bild på hur tabellen för tomat ser ut.

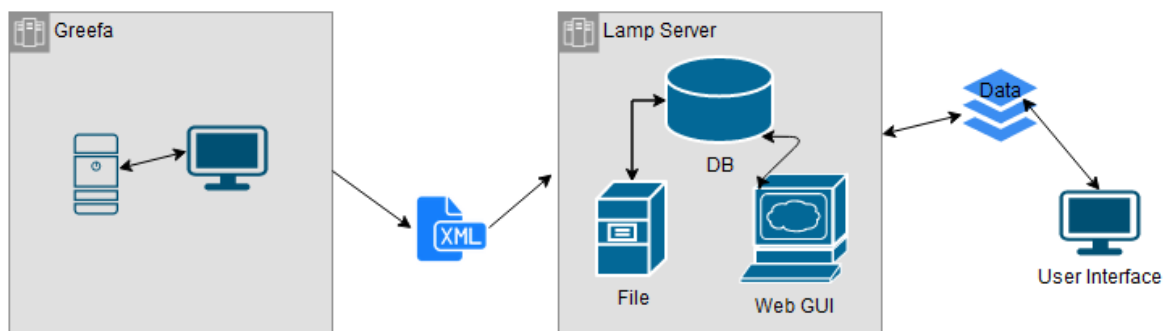
#### 4.5 Import av filer

Eftersom det behövdes automatisk import av XML-filer till databasen så skapades ett PHP-skript som läser in alla XML-filer från en mapp och sparar till tabellen. Detta PHP-skript körs automatiskt varje minut av ett system som finns inbyggt i Ubuntu som heter crontab och i denna funktion kan man ställa in vad som skall köras och när. På linjerna skapades ett

”.bat” script som överför automatiskt varje fil så fort den känner av att en ny fil har skapats till specifik mapp, och den läser aldrig över filer som är äldre än en dag. Eftersom vi inte behöver spara dessa XML-filer i vår databas så tas varje enskild fil bort så fort den har lästs in av PHP-skriptet.

## 4.6 Webbapplikationen

Själva webbapplikationen gjordes att man kan fylla på databasen med den data som inte kan fås via XML-filer från maskinen, så som personalantal, driftstopp med mera. Varje linje bestod av tre olika PHP-sidor, en primär sida, redigeringsida och “visa” sida som man har länkat till via olika knappar.



Figur 11. Hur systemet kommer att fungera.

## 4.7 Innehåll i webbsidorna

Då nya data importerats till databasen så visas det på primär sidan, ett nytt fält som skall fyllas på med data. Sidan visar olika status med färger, där status ett är importerat med ljusblå färg, status två är sparad orange färg, och status tre är klart med grön färg. Då nya data har importerats visas status ett och man går då in på redigera knappen för att mata in data och de fält som inte behövs matas in är låsta, medan de fält som måste matas in är det krav på att mata in före man sparar. För att spara fanns det två knappar, en spara och en överför knapp, då man överför så kan man inte redigera fältet något mera och man återgår till första sidan för den linje man matar data in för.

De flesta fälten är vanliga textboxar med olika attribut och en del rullgardinsmenyer för att få mera val. Driftstoppen hämtas från databasen och de blir tillagda vid varje import av nya data från XML.

För att hämta data från maskinen som skulle lagras på databasen så görs detta via en PHP-skript. Man läser en fil i taget och tar ut de taggar man behöver och vill använda, dessa eventuellt formateras att databasen kan hantera informationen och sparas.

```

1 $odlarNummer = trim($xml->Odlarnummer);
2 $leveransDatum = date_create_from_format('d-m-Y', $xml->Leveransdatum);
3 $leveransDatum = $leveransDatum->format("Y-m-d");
4 $sorteringsdatum = date_create_from_format('d-m-Y', $xml->Sorteringsdatum);
5 $sorteringsdatum = $sorteringsdatum->format("Y-m-d");
6 $forstaKlassVikt = str_replace(".", "", trim($xml->IKlassKg));
7 $andraKlassVikt = str_replace(".", "", trim($xml->IIKlassKg));
8 $avfallKg = str_replace(".", "", trim($xml->AvfallKg));
9 $totalVikt = str_replace(".", "", trim($xml->TotaltVikt));
10 $rodaTomater = str_replace(",", ".", trim($xml->RödaTomater));
11 $gronaTomater = str_replace(",", ".", trim($xml->GrönaTomater));

```

**Figur 12 Bild över halva importeringskoden.**

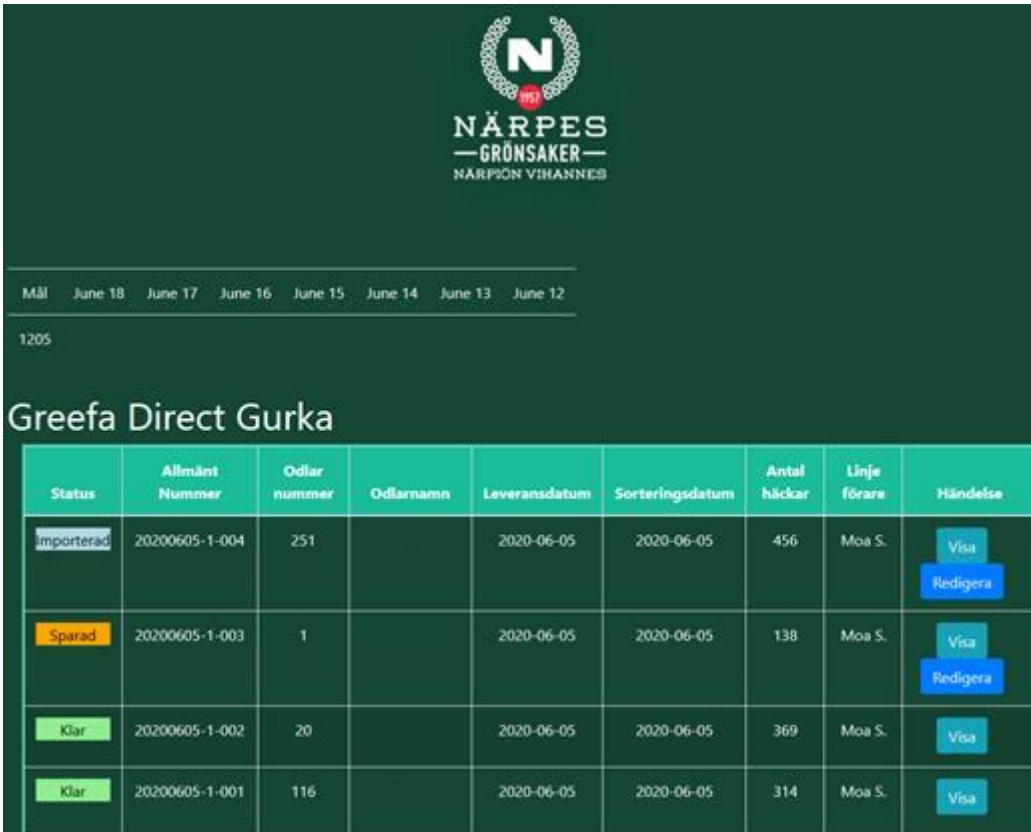
Då de taggar som man skall hämta information från kunde ibland ha extra utrymme tog man bort detta med funktionen ”trim” i importen av XML-filerna. Vid importen behövdes datum formateras med funktionen ”date\_create\_from\_format” och formatet behövdes ange eftersom det inte vara samma på alla fält. Datumet sparades i databasen enligt formatet åååå-mm-dd. Funktionen ”str\_replace” användes eftersom man behövde ta bort punkt i vissa fall, och i andra ändra kommatecken till punkt.

## 5 Resultat

Slutresultatet blev en webbapplikation som kan hantera information från maskinen till slutanvändaren. Överföringen från själva maskinen till servern sköts automatiskt med ett batch skript, det enda som användaren måste komma ihåg är att detta skript måste vara igång annars kommer inget upp på webbsidan. XML-filerna skapas av Greefa-programmet som sparar all information som behövs i olika taggar, och efter att filen har skapats kan importen hända. Efter import av filerna till servern så hanteras de importerade filerna automatiskt med ett PHP-skript som kör in dem till databasen och raderar själva filen från servern. Då allt detta har utförts så kan användaren redigera det nya fältet som har kommit fram, då användaren är nöjda med det de har fyllt i så kan de använda knappen spara eller knappen överföra.

I första layouten som gjordes var det tänkt att driftstopp skulle vara i egen tabell och man skulle använda rullgardinsmeny och textruta som är formaterad att bara ta emot nummer, där man kunde välja vilket driftstopp man ville ha och lägga till minuter. För att lägga till mera rullgardinsmenyer så skulle man göra via en knapp som gjorde att nya fält lades till. Men kunden ville ha allt i en enda tabell och alla driftstopp skall vara i separata textrutor som skall alltid vara synliga. Detta gör att det kan bli ganska mycket information som inte alltid används men måste läggas till. Eftersom varje driftstopp har ett namn, antal minuter och kommentars fält, måste man lägga till namnet på driftstoppet och antal minuter till noll vid varje import av XML-filer och siffran noll måste läggas till eftersom fältet är formaterat som heltal.

Webbsidan är gjord minimalistiskt så att det är enkelt att använda och att användaren skall ha det enkelt att hitta det de skall göra. Första sidan visar de importerade partier, vilka partier man har ändrat och sparat bara och vilka som är slutförda. Sidan har också ett dynamiskt skript som räknar ut totala kilo producerade per dag. Det finns tre statusar dessa kontrollerar om knappen redigera skall visas eller inte. Varje parti som importeras får status ett, då man går inte och redigerar får partiet status två och då man klickar på knappen överför så får den status tre och detta gömmer knappen redigera.



The screenshot shows the Greefa Direct Gurka interface. At the top, there is a logo for NÄRPES GRÖNSAKER NÄRPES VIBANNES. Below the logo, there is a navigation bar with dates from June 12 to June 18. The main content area is titled 'Greefa Direct Gurka' and contains a table with the following data:

Status	Allmänt Nummer	Odlar nummer	Odlarnamn	Leveransdatum	Sorteringsdatum	Antal häckar	Linje förare	Händelse
Importerad	20200605-1-004	251		2020-06-05	2020-06-05	456	Moa S.	Visa Redigera
Sparad	20200605-1-003	1		2020-06-05	2020-06-05	138	Moa S.	Visa Redigera
Klar	20200605-1-002	20		2020-06-05	2020-06-05	369	Moa S.	Visa
Klar	20200605-1-001	116		2020-06-05	2020-06-05	314	Moa S.	Visa

Figur 13. Bild på utseendet på första sidan av Gurka.

Då de väljer att redigera så kommer de till en sida där man kan fylla på mera med information. Det finns alla de fält som fanns i Excel-dokumentet som de använde förr i tiden och ännu flera saker att välja mellan och fylla i. Vissa fält är låsta då dessa kommer från XML-filen och man vill inte att användaren skall redigera dessa, som t.ex. allmänt nummer fältet med mera. Vissa fält måste fyllas i för att man skall kunna spara eller överföra sidan, så som antal personer på linjen, partifördelning om partiet packades i helhet eller om man hade delat upp och vilken del av detta som packades, start och stopptiden. Start och stopptiden fanns inte i början i XML-filerna men denna funktion fanns i maskinen, då fick Greefa lägga till denna tag i XML-filerna för att man inte skulle behöva skriva upp när varje parti började och slutade på datorn eller på ett papper.

ID-nummer	20200605-1-003
Odlarnummer	1
Odlarnamn	
Produktkod	2201
Leveransdatum	2020 - 06 - 05
Antal häckar	138
Linjeförare	Moa S.
Personer på linjen	13
Partifördelning	<input type="radio"/> Partiet i helhet packat <input checked="" type="radio"/> Första delen av ett parti <input type="radio"/> Fortsättning på ett parti
Sorteringsdatum	2020 - 06 - 05
Sortering starttid	12:48:00
Sortering stopptid	13:14:04
Hastighet på maskinen	<input checked="" type="radio"/> 80 % <input type="radio"/> 85 % <input type="radio"/> 90 % <input type="radio"/> 95 % <input type="radio"/> 100 %
Kaffepauser, antal	0
Matpauser, ange längd	0
Annan paus (minuter)	0
Orsak för annan paus	

**Figur 14.** Bild över sidan som visas då man klickar redigera-knappen.

All denna information som finns i databasen ville de kunna göra något av, en Excel-exportsida skapades för varje linje eftersom varje linje har en egen tabell. Där kan man välja från och till vilket datum man ville ha partiinformation om.

## 6 Diskussion

Ingenjörarbetet påbörjades med att analysera nuvarande läget och reda ut vilken information som skall återanvändas och vilken som måste skapas helt ny. De hade från förr redan köpt in liknande webbapplikationer detta gjorde saken lite enklare för att man kunde se vilken information och hur användaren upplevde att den applikationen fungerade. Efter att ha gjort analysen och sett hur de använde Excel-dokumentet, började jag med att skapa en första design av websidan för att hantera data från databasen. Själv hade man grundkunskaper i PHP och JS det gjorde att jag fick fördjupa mig mera för att få det att fungera som de ska.

Installation av Ubuntu server gick bra men det uppstod vissa problem så som rättigheter till mappar. Fick också reda ut hur man skall få mappar delade till nätverket och kunna hantera filer som kopieras över, till detta fick det bli Samba servertillägg till Ubuntu och man fick dela ut rättigheter till par mappar till alla. Kopieringen av filerna från servern i produktionshallen till Ubuntu servern gick bra eftersom de hade likande sätt men jag valde robocopy som är en modernare version av xcopy, denna metod programmerades in i batch skript som skulle föra över filerna till den nätverksdelade mappen.

Skapade websidan på egen dator allt verkade fungera och efter att fått godkännande laddade jag upp filerna till servern och skapade databasen då började det riktiga testet. Användaren fyllde i både Excel filen och websidan tills allt verkade bra för att de skulle kunna hoppa över till websidan bara. Första versionen gjordes bara för den så kallade tomatsidan och sedan klonades över till de andra linjerna där applikationen skulle användas med vissa ändringar. Efter att ha klonat och skapat för de andra maskinerna detta skulle användas på blev projektet färdigt. Efter färdigt projekt så har detta lett till att man har fått projektanställning för att skapa en fortsättning i form av en administrativ sida och uppdatera detta befintliga projekt med nya funktioner.



## 7 Referenser

- Artūras, B. (den 07 Juni 2020). *Hostinger*. Hämtat från <https://www.hostinger.com/tutorials/what-is-css>
- Berezhnoi, R. (den 05 Juni 2020). *F5 Studio*. Hämtat från <https://f5-studio.com/articles/cms-or-frameworks/>
- Bilqis, A. (den 05 Juni 2020). *Hostinger*. Hämtat från <https://www.hostinger.com/tutorials/what-is-php/>
- Debian*. (den 7 Maj 2020). Hämtat från <https://www.debian.org/releases/stable/arm64/ch01s02.sv.html>
- Domantas, G. (den 07 Juni 2020). *Hostinger*. Hämtat från <https://www.hostinger.com/tutorials/what-is-javascript>
- Herawan, D. (den 05 Juni 2020). *Hostinger*. Hämtat från <https://www.hostinger.com/tutorials/what-is-mysql>
- Microsoft*. (den 07 Maj 2020). Hämtat från <https://azure.microsoft.com/sv-se/overview/what-is-a-virtual-machine/>
- Oxford Web Studio*. (den 02 Juli 2020). Hämtat från <https://www.oxfordwebstudio.com/sv/vet-du/vad-ar-html>
- Sörhannus, T. (2007). *Närpes Grönsaker 50 år*.
- Ubuntu*. (den 10 Juli 2020). Hämtat från <https://ubuntu.com/about>
- Wordpress*. (den 7 Maj 2020). Hämtat från <https://sv.wordpress.org/>