09-12-2020

Patrick Lodahl Gravengaard - 3.Z

Herningsholm erhversskole og gymnasier

Objekter i Flask

Programmering B

Indhold

[Objekter i Flask 2](#_Toc58446717)

[Beskrivelse af ide 2](#_Toc58446718)

[Link til programmet 2](#_Toc58446719)

[Prioriteret liste af krav 2](#_Toc58446720)

[Klassediagram 3](#_Toc58446721)

[Skærmbilleder af websider 4](#_Toc58446722)

[Forside 4](#_Toc58446723)

[Opretning af spiller 5](#_Toc58446724)

[Slet spiller 5](#_Toc58446725)

[Redigering af spiller 6](#_Toc58446726)

[Forklaring af udvalgt kode for henholdsvis model, view og controller. 6](#_Toc58446727)

[Model 6](#_Toc58446728)

[View 7](#_Toc58446729)

[Controller 8](#_Toc58446730)

[Bilag 9](#_Toc58446731)

[Player.py 9](#_Toc58446732)

[app.py 10](#_Toc58446733)

[index.html 17](#_Toc58446734)

[Update.html 19](#_Toc58446735)

[Create.html 20](#_Toc58446736)

[Delete.html 20](#_Toc58446737)

[Createattacker.html 21](#_Toc58446738)

[Createdefender.html 21](#_Toc58446739)

[Createmidfielder.html 22](#_Toc58446740)

[Creategoalkeeper.html 22](#_Toc58446741)

# Objekter i Flask

# Beskrivelse af ide

Ideen til mit program, er en side hvor man kan se forskellige spillers data. Så en hjemmeside beregnet til scouting af fodboldspillere. Man skal kunne se spillerens alder, deres navn, deres hold og deres præstationer - mål, bolde spillet osv.

# Link til programmet

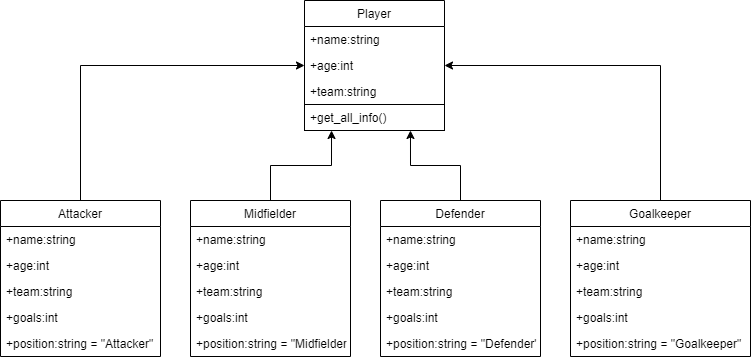
<http://flaskafleveringprogram.pythonanywhere.com/>

# Prioriteret liste af krav

* Opretning af spiller
* Ændring af spiller (Tilføj mål etc.)
* Mulighed for at fjerne en spiller
* Visualisering med evt. grafer
* Sammenligning af spillere
* Rigtige data

Dette er kravene som jeg har opstillet. Som der kan ses, har jeg fået lavet det meste. Jeg har ikke fået lavet de sidste pga. tidsmangel og mangel på mere data, som kunne bruges. Men de krav som er selve skelettet af programmet er udført, hvor de manglende krav er tilføjelser.

# Klassediagram



Jeg har opsat et dette klassediagram, som viser at Player er superklassen, hvor Attacker, Midfielder, Defender og Goalkeeper er underklasserne til superklassen Player. Der er indsat alle parameteren alle klasser kan tage imod og deres funktioner, hvis de har nogle.

# Skærmbilleder af websider

## Forside



Forsiden er der hvor man ser alt informationen om spilleren, hvilket så også er den vigtigste side. Her kan man se spillerne delt op i om de er en målmand, en forsvar, midtbane eller angriber. I hver af disse kategorier, kan man så se spillerne. Hvor de hver har et ID, et navn, en alder, et hold og en parameter som passer til deres rolle, som f.eks. mål, ved en angriber. I toppen kan man så vælge om man vil oprette, slette eller redigere en spiller.

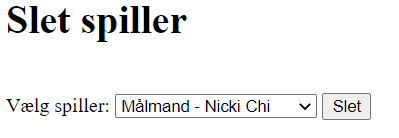
## Opretning af spiller





Når man skal oprette en spiller, kommer man først ind til siden øverst, hvor man skal vælge om man vil oprette en målmand, en forsvar spiller, en midtbane spiller eller en angriber. Efter man har valgt en position, kommer man ind til en side som f.eks. kan ses over, hvor man så kan indsætte alle værdierne til opretning af spilleren.

## Slet spiller



Når man skal slette en spiller, kommer man ind på siden, som set ovenfor. Her vælger man bare hvilken spiller man vil slette ud fra listen af spillere og klikker på slet knappen.

## Redigering af spiller



Til redigering af en spiller, vælger man igen hvilken spiller man vil oprette og man indsætter herefter de nye værdier for spilleren og klikker på rediger. Hvis man nu kun f.eks. skal ændre alderen, indsætter man bare den nye alder og klikker rediger. De gamle værdier vil så stadig være de samme, men alderen bliver så kun opdateret.

# Forklaring af udvalgt kode for henholdsvis model, view og controller.

## Model

*# Klassen player*

class Player():

*# Definere \_\_init\_\_ med parametrene name, age og team*

    def \_\_init\_\_(self, name, age, team):

        self.name = name

        self.age = age

        self.team = team

*# Funktionen get\_all\_info som retunere variablerne og en beskrivelse.*

    def get\_all\_info(self):

*return* f"Spilleren hedder {self.name}, han er {self.age} år gammel. Han spiller for {self.team}"

*# Klassen Attacker, som er har superklassen Player*

class Attacker(Player):

*# Sætter id\_generator til at starte ved 1*

    id\_generator = itertools.count(1)

*# Definere \_\_init\_\_ med parametrene name, age, team, goals, position="attacker"*

    def \_\_init\_\_(self, name, age, team, goals, position="Attacker"):

*# Inkludere superklassen, hvor der skal bruges parameterne name, age og*

*# team*

        super().\_\_init\_\_(name, age, team)

*# Sætter id til den næste, altså 1,2,3,4 osv*

        self.id = next(self.id\_generator)

*# Sætter goals lig med inputtet for goals*

        self.goals = goals

I model er alle klasserne, (fulde version kan ses i bilag), hvor alt dataene bliver oprettet. Model håndterer alt datene og sender det ud til controller. I koden bliver der først oprettet en klasse ved navn Player, hvor den har parametrene name, age og team. Den har desuden funktionen get\_all\_info() som returnerer variablerne og en sammenhængende tekst til dem. I den næste klasse, som er en af underklasserne til Player, som hedder Attacker. I Attacker er der en variabel kaldet id\_generator, som bruges til at give et unikt ID til hver spiller. Attacker tager imod parametrene name, age, team, goals, og position. Den bruger desuden parametrene name, age og team fra superklassen.

## View

*<!-- Målmand -->*

    <h3>Målemænd</h3>

    <table>

      <tr>

        <th>ID</th>

        <th>Navn</th>

        <th>Alder</th>

        <th>Hold</th>

        <th>Redninger</th>

      </tr>

      {# Viewet viser spillerne i hver sin række. #}

      {% for goalkeeper in goalkeepers %}

      <tr>

        <td>{{ goalkeeper.id }}</td>

        <td>{{ goalkeeper.name }}</td>

        <td>{{ goalkeeper.age }}</td>

        <td>{{ goalkeeper.team }}</td>

        <td>{{ goalkeeper.saves }}</td>

      </tr>

      {% endfor %}

    </table>

I view er alt det grafiske, som i dette eksempel er html, hvor der er et samarbejde mellem controller og view. Da controller sender de variabler som view skal bruge for at vise dem. I selve udsnittet af ”koden”, er der øverst en h3, som er lille overskrift, som viser at det er målmændene denne sektion handler om. Herefter er der en tabel hvor der er ID, Navn, Alder, Hold og Redninger og under vises der ved hjælp af Jinja målmændene og deres ID, navn osv. (Jinja går igennem hver værdi i goalkeepers listen.)

## Controller

*# Siden /delete*

@app.route('/delete', methods=['POST', 'GET'])

def delete():

*# Hvis der postes noget data*

*if* request.method == 'POST':

*# Opretter variablen choice, som tager fat i værdierne fra inputtet fra*

*# deleteplayer*

        choice = request.form.get('deleteplayer')

*# Tjekker om den første karakter i choice er lig med det givne bogstav*

*# Hvis den er det, laves der er en varibel kaldet get\_index som tager alt*

*# det efter det*

*# Første bogstav. Herefter tjekker den igennem listen og finder der hvor*

*# den spiller man skal*

*# Findes id og fjerner derefter den spiller.*

*if* choice[0] == "g":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(goalkeepers)):

*if* goalkeepers[x].id == int(get\_index):

                    index = x

            goalkeepers.pop(int(index))

I controller regner den alt den information den får for model og passere det herefter videre til view, som så viser det. Så controller er et logiklager hvor alt bliver regnet. I kode udsnittet, kan der ses at det er tilknyttet til siden /delete. Det første den gør, er at tjekke om der bliver postet noget data. Hvis der bliver postet noget data, bliver variablen choice sat lig med værdien fra inputtet, deleteplayer. Herefter tjekkes der så om den første karakter i choice er lig med g. Hvis den er dette, laver den en variabel kaldet get\_index som er lig med værdierne efter den første karakter. Herefter sættes en variabel index lig med 0. Herefter går den igennem alle værdier i listen goalkeepers og tjekker om id’et passer med id’et i get\_index. Hvis et id passer med get\_index bliver index sat lig med pladsen id’et var fundet i. Til sidst fjernes denne spiller.

# Bilag

## Player.py

*# Importer itertools, som f.eks. kan bruges til at lave ID'er*

*import* itertools

*# Klassen player*

class Player():

*# Definere \_\_init\_\_ med parameterene name, age og team*

    def \_\_init\_\_(self, name, age, team):

        self.name = name

        self.age = age

        self.team = team

*# Funktionen get\_all\_info som retunere variablerne og en beskrivelse.*

    def get\_all\_info(self):

*return* f"Spilleren hedder {self.name}, han er {self.age} år gammel. Han spiller for {self.team}"

*# Klassen Attacker, som er har superklassen Player*

class Attacker(Player):

*# Sætter id\_generator til at starte ved 1*

    id\_generator = itertools.count(1)

*# Definere \_\_init\_\_ med parameterene name, age, team, goals, position="attacker"*

    def \_\_init\_\_(self, name, age, team, goals, position="Attacker"):

*# Inkludere superklassen, hvor der skal bruges parameterne name, age og team*

        super().\_\_init\_\_(name, age, team)

*# Sætter id til den næste, altså 1,2,3,4 osv*

        self.id = next(self.id\_generator)

*# Sætter goals lig med inputtet for goals*

        self.goals = goals

*# Klassen Midfielder, som har superklassen Player*

class Midfielder(Player):

*# Sætter id\_generator til at starte ved 1*

    id\_generator = itertools.count(1)

*# Definere \_\_init\_\_ med parameterene name, age, team, passes, position="Midfielder"*

    def \_\_init\_\_(self, name, age, team, passes, position="Midfielder"):

*# Inkludere superklassen, hvor der skal bruges parameterne name, age og team*

        super().\_\_init\_\_(name, age, team)

*# Sætter passes lig med inputtet for passes*

        self.passes = passes

*# Sætter id til den næste, altså 1,2,3,4 osv*

        self.id = next(self.id\_generator)

*# Klassen Defender, som har superklassen Player*

class Defender(Player):

*# Sætter id\_generator til at starte ved 1*

    id\_generator = itertools.count(1)

*# Definere \_\_init\_\_ med parameterene name, age, team, blocked, position="Defender"*

    def \_\_init\_\_(self, name, age, team, blocked, position="Defender"):

*# Inkludere superklassen, hvor der skal bruges parameterne name, age og team*

        super().\_\_init\_\_(name, age, team)

*# Sætter blocked lig med inputtet for blocked*

        self.blocked = blocked

*# Sætter id til den næste, altså 1,2,3,4 osv*

        self.id = next(self.id\_generator)

*# Klassen Goalkeeper, som har superklassen Player*

class Goalkeeper(Player):

*# Sætter id\_generator til at starte ved 1*

    id\_generator = itertools.count(1)

*# Definere \_\_init\_\_ med parameterene name, age, team, position, position="Goalkeeper"*

    def \_\_init\_\_(self, name, age, team, saves, position="Goalkeeper"):

*# Inkludere superklassen, hvor der skal bruges parameterne name, age og team*

        super().\_\_init\_\_(name, age, team)

*# Sætter saves lig med inputtet for saves*

        self.saves = saves

*# Sætter id til den næste, altså 1,2,3,4 osv*

        self.id = next(self.id\_generator)

## app.py

*# Importering af bibloteker og filer*

*from* flask *import* Flask, render\_template, request

*from* player *import* Attacker, Goalkeeper, Defender, Midfielder

app = Flask(\_\_name\_\_)

*# Lister som opbevare dataene*

goalkeepers = [Goalkeeper("Nicki Chi", 23, "Trolp FC", 18)]

defenders = [Defender("Alexander Jul", 25, "Tripa FC", 32)]

midfielders = [Midfielder("Adam Sul", 19, "Okaype FC", 200)]

attackers = [Attacker("Jacob Lund", 21, "Yeppep FC", 8)]

*# Startsiden/index*

@app.route('/')

def index():

*# Retunere index.html, med de forskellige lister*

*return* render\_template('index.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

*# Siden /create*

@app.route('/create', methods=['POST', 'GET'])

def create():

*# Hvis der postes noget data*

*if* request.method == 'POST':

*# Opretter variablen choice, som tager fat i værdierne fra inputtet fra CreateType*

        choice = request.form.get('CreateType')

*# Tjekker om inputtet er lig med "Attacker", "Defender" osv.*

*# Den retunere derefter siden som passer med inputtet*

*if* choice == "Attacker":

*return* render\_template('createattacker.html', page=createattacker)

*elif* choice == "Defender":

*return* render\_template('createdefender.html', page=createdefender)

*elif* choice == "Midfielder":

*return* render\_template('createmidfielder.html', page=createmidfielder)

*elif* choice == "Goalkeeper":

*return* render\_template('creategoalkeeper.html', page=creategoalkeeper)

*# Hvis der ikke bliver postet noget data*

*# Så retunere den create.html*

*else*:

*return* render\_template('create.html')

*# Siden /createattacker*

@app.route('/createattacker', methods=['POST', 'GET'])

def createattacker():

*# Hvis der postes noget data*

*# Sæt de forskellige parameter som klassen attacker bruger*

*# Lig med værdierne i inputsne og retunere index.html*

*if* request.method == 'POST':

        name = request.form['name']

        age = request.form['age']

        team = request.form['team']

        goals = request.form['goals']

        attackers.append(Attacker(name, int(age), team, int(goals)))

*return* render\_template('index.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

*# Hvis der ikke postes noget data*

*# Retunere createattacker.html*

*else*:

*return* render\_template('createattacker.html')

*# Siden /creategoalkeeper*

@app.route('/creategoalkeeper', methods=['POST', 'GET'])

def creategoalkeeper():

*# Hvis der postes noget data*

*# Sæt de forskellige parameter som klassen goalkeeper bruger*

*# Lig med værdierne i inputsne og retunere index.html*

*if* request.method == 'POST':

        name = request.form['name']

        age = request.form['age']

        team = request.form['team']

        saves = request.form['saves']

        goalkeepers.append(Goalkeeper(name, int(age), team, int(saves)))

*return* render\_template('index.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

*# Hvis der ikke postes noget data*

*# Retunere creategoalkeeper.html*

*else*:

*return* render\_template('creategoalkeeper.html')

*# Siden /createdefender*

@app.route('/createdefender', methods=['POST', 'GET'])

def createdefender():

*# Hvis der postes noget data*

*# Sæt de forskellige parameter som klassen defender bruger*

*# Lig med værdierne i inputsne og retunere index.html*

*if* request.method == 'POST':

        name = request.form['name']

        age = request.form['age']

        team = request.form['team']

        blocked = request.form['blocked']

        defenders.append(Defender(name, int(age), team, int(blocked)))

*return* render\_template('index.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

*# Hvis der ikke postes noget data*

*# Retunere createdefender.html*

*else*:

*return* render\_template('createdefender.html')

*# Siden /createmidfielder*

@app.route('/createmidfielder', methods=['POST', 'GET'])

def createmidfielder():

*# Hvis der postes noget data*

*# Sæt de forskellige parameter som klassen midfielder bruger*

*# Lig med værdierne i inputsne og retunere index.html*

*if* request.method == 'POST':

        name = request.form['name']

        age = request.form['age']

        team = request.form['team']

        passes = request.form['passes']

        midfielders.append(Midfielder(name, int(age), team, int(passes)))

*return* render\_template('index.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

*# Hvis der ikke postes noget data*

*# Retunere createmidfielder.html*

*else*:

*return* render\_template('createmidfielder.html')

*# Siden /delete*

@app.route('/delete', methods=['POST', 'GET'])

def delete():

*# Hvis der postes noget data*

*if* request.method == 'POST':

*# Opretter variablen choice, som tager fat i værdierne fra inputtet fra deleteplayer*

        choice = request.form.get('deleteplayer')

*# Tjekker om den første karakter i choice er lig med det given bogstav*

*# Hvis den er det laves der er en varibel kaldet get\_index som tager alt det efter det*

*# Første bogstav. Herefter tjekker den igennem listen og finder der hvor den spiller man skal*

*# Findes id og fjerner derefter den spiller.*

*if* choice[0] == "g":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(goalkeepers)):

*if* goalkeepers[x].id == int(get\_index):

                    index = x

            print(index)

            goalkeepers.pop(int(index))

*elif* choice[0] == "d":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(defenders)):

*if* defenders[x].id == int(get\_index):

                    index = x

            defenders.pop(int(index))

*elif* choice[0] == "m":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(midfielders)):

*if* midfielders[x].id == int(get\_index):

                    index = x

            midfielders.pop(int(index))

*elif* choice[0] == "a":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(attackers)):

*if* attackers[x].id == int(get\_index):

                    index = x

            attackers.pop(int(index))

*return* render\_template('index.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

*# Hvis der ikke postes noget data*

*# Retunere delete.html*

*else*:

*return* render\_template('delete.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

@app.route('/update', methods=['POST', 'GET'])

def update():

*# Hvis der postes noget data*

*if* request.method == 'POST':

*# Opretter variablen choice, som tager fat i værdierne fra inputtet fra updateplayer*

        choice = request.form.get('updateplayer')

*# Laver variabler som henter værdierne fra deres tilknutne inputs*

        update\_name = request.form['update\_name']

        update\_age = request.form['update\_age']

        update\_team = request.form['update\_team']

        update\_value = request.form['update\_value']

*# Tjekker om den første karakter i choice er lig med det given bogstav*

*# Hvis den er det laves der er en varibel kaldet get\_index som tager alt det efter det*

*# Første bogstav. Herefter tjekker den igennem listen og finder der hvor den spiller man skal*

*# Findes id. Der tjekkes der efter om der mangler noget input i nogle felter, og hvis det mangler noget*

*# Sættes det bare lig med den gamle værdi, men hvis der er en ny, sættes det til den nye værdi.*

*if* choice[0] == "g":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(goalkeepers)):

*if* goalkeepers[x].id == int(get\_index):

                    index = x

*if* update\_name == '':

                goalkeepers[index].name = goalkeepers[index].name

*else*:

                goalkeepers[index].name = update\_name

*if* update\_age == '':

                goalkeepers[index].age = goalkeepers[index].age

*else*:

                goalkeepers[index].age = update\_age

*if* update\_team == '':

                goalkeepers[index].team = goalkeepers[index].team

*else*:

                goalkeepers[index].team = update\_team

*if* update\_value == '':

                goalkeepers[index].saves = goalkeepers[index].saves

*else*:

                goalkeepers[index].saves = update\_value

*elif* choice[0] == "d":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(defenders)):

*if* defenders[x].id == int(get\_index):

                    index = x

*if* update\_name == '':

                defenders[index].name = defenders[index].name

*else*:

                defenders[index].name = update\_name

*if* update\_age == '':

                defenders[index].age = defenders[index].age

*else*:

                defenders[index].age = update\_age

*if* update\_team == '':

                defenders[index].team = defenders[index].team

*else*:

                defenders[index].team = update\_team

*if* update\_value == '':

                defenders[index].blocked = defenders[index].blocked

*else*:

                defenders[index].blocked = defenders

*elif* choice[0] == "m":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(midfielders)):

*if* midfielders[x].id == int(get\_index):

                    index = x

*if* update\_name == '':

                midfielders[index].name = midfielders[index].name

*else*:

                midfielders[index].name = update\_name

*if* update\_age == '':

                midfielders[index].age = midfielders[index].age

*else*:

                midfielders[index].age = update\_age

*if* update\_team == '':

                midfielders[index].team = midfielders[index].team

*else*:

                midfielders[index].team = update\_team

*if* update\_value == '':

                midfielders[index].passes = midfielders[index].passes

*else*:

                midfielders[index].passes = update\_value

*elif* choice[0] == "a":

            get\_index = choice[1:]

            index = 0

*for* x *in* range(len(attackers)):

*if* attackers[x].id == int(get\_index):

                    index = x

*if* update\_name == '':

                attackers[index].name = attackers[index].name

*else*:

                attackers[index].name = update\_name

*if* update\_age == '':

                attackers[index].age = attackers[index].age

*else*:

                attackers[index].age = update\_age

*if* update\_team == '':

                attackers[index].team = attackers[index].team

*else*:

                attackers[index].team = update\_team

*if* update\_value == '':

                attackers[index].goals = attackers[index].goals

*else*:

                attackers[index].goals = update\_value

*return* render\_template('index.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

*# Hvis der ikke postes noget data*

*# Retunere delete.html*

*else*:

*return* render\_template('update.html', goalkeepers=goalkeepers, defenders=defenders, midfielders=midfielders, attackers=attackers)

app.run(debug=True, host='0.0.0.0', port=81)

## index.html

<html>

  <head>

    <link rel="stylesheet" href="/static/style.css" />

  </head>

  <body>

    <a href="/create">Opret spiller</a>

    <a href="/delete">Slet spiller</a>

    <a href="/update">Rediger spiller</a>

    <h1>Spillere</h1>

*<!-- Målmand -->*

    <h3>Målemænd</h3>

    <table>

      <tr>

        <th>ID</th>

        <th>Navn</th>

        <th>Alder</th>

        <th>Hold</th>

        <th>Redninger</th>

      </tr>

      {# Viewet viser spillerne i hver sin række. #}

      {% for goalkeeper in goalkeepers %}

      <tr>

        <td>{{ goalkeeper.id }}</td>

        <td>{{ goalkeeper.name }}</td>

        <td>{{ goalkeeper.age }}</td>

        <td>{{ goalkeeper.team }}</td>

        <td>{{ goalkeeper.saves }}</td>

      </tr>

      {% endfor %}

    </table>

    <br />

*<!-- Forsvar -->*

    <h3>Forsvar</h3>

    <table>

      <tr>

        <th>ID</th>

        <th>Navn</th>

        <th>Alder</th>

        <th>Hold</th>

        <th>Bolde blokeret</th>

      </tr>

      {# Viewet viser spillerne i hver sin række. #}

      {% for defender in defenders %}

      <tr>

        <td>{{ defender.id }}</td>

        <td>{{ defender.name }}</td>

        <td>{{ defender.age }}</td>

        <td>{{ defender.team }}</td>

        <td>{{ defender.blocked }}</td>

      </tr>

      {% endfor %}

    </table>

    <br />

*<!-- Midtbane -->*

    <h3>Midtbane</h3>

    <table>

      <tr>

        <th>ID</th>

        <th>Navn</th>

        <th>Alder</th>

        <th>Hold</th>

        <th>Afleveringer</th>

      </tr>

      {# Viewet viser spillerne i hver sin række. #}

      {% for midfielder in midfielders %}

      <tr>

        <td>{{ midfielder.id }}</td>

        <td>{{ midfielder.name }}</td>

        <td>{{ midfielder.age }}</td>

        <td>{{ midfielder.team }}</td>

        <td>{{ midfielder.passes }}</td>

      </tr>

      {% endfor %}

    </table>

    <br />

*<!-- Angriber -->*

    <h3>Angribere</h3>

    <table>

      <tr>

        <th>ID</th>

        <th>Navn</th>

        <th>Alder</th>

        <th>Hold</th>

        <th>Mål</th>

      </tr>

      {# Viewet viser spillerne i hver sin række. #}

      {% for attacker in attackers %}

      <tr>

        <td>{{ attacker.id }}</td>

        <td>{{ attacker.name }}</td>

        <td>{{ attacker.age }}</td>

        <td>{{ attacker.team }}</td>

        <td>{{ attacker.goals }}</td>

      </tr>

      {% endfor %}

    </table>

  </body>

</html>

## Update.html

<html>

<head>

<link rel="stylesheet" href='/static/style.css' />

</head>

<body>

<h1>Rediger spiller</h1>

<br>

<form method="post" action="/update">

    <label for="update">Vælg spiller:</label>

    <select id="update" name="updateplayer" size="1">

      {% for goalkeeper in goalkeepers %}

      <option name="updateplayer" value="g{{ goalkeeper.id }}">Målmand - {{ goalkeeper.name }}</option>

      {% endfor %}

      {% for defender in defenders %}

      <option name="updateplayer" value="d{{ defender.id }}">Forsvar - {{ defender.name }}</option>

      {% endfor %}

      {% for midfielder in midfielders %}

      <option name="updateplayer" value="m{{ midfielder.id }}">Midtbane - {{ midfielder.name }}</option>

      {% endfor %}

      {% for attacker in attackers%}

      <option name="updateplayer" value="a{{ attacker.id }}">Angriber - {{ attacker.name }}</option>

      {% endfor %}

    </select>

    <br>

    <p>Indsæt ny værdi for redninger/blokeret/afleveringer/mål</p>

    <input type="text" name="update\_name" placeholder="Indsæt nyt navn">

    <input type="number" name="update\_age" placeholder="Indsæt ny alder">

    <input type="text" name="update\_team" placeholder="Indsæt nyt hold">

    <input type="number" name="update\_value" placeholder="Indsæt ny værdi">

    <input type="submit" value="Rediger">

  </form>

</body>

</html>

## Create.html

</form>

</body>

</html>

<!doctype html>

<html>

<head>

<title>Opret spiller</title>

</head>

<body>

<h1>Opret spiller</h1>

<form method="post">

<input type="checkbox" name="CreateType" value="Goalkeeper"> Goalkeeper <br>

<input type="checkbox" name="CreateType" value="Defender"> Defender <br>

<input type="checkbox" name="CreateType" value="Midfielder"> Midfielder <br>

<input type="checkbox" name="CreateType" value="Attacker"> Attacker <br>

<input type="submit" value="Opret">

</form>

</body>

</html>

## Delete.html

<html>

<head>

<link rel="stylesheet" href='/static/style.css' />

</head>

<body>

<h1>Slet spiller</h1>

<br>

<form method="post" action="/delete">

  <label for="delete">Vælg spiller:</label>

  <select id="delete" name="deleteplayer" size="1">

    {% for goalkeeper in goalkeepers %}

    <option name="deleteplayer" value="g{{ goalkeeper.id }}">Målmand - {{ goalkeeper.name }}</option>

    {% endfor %}

    {% for defender in defenders %}

    <option name="deleteplayer" value="d{{ defender.id }}">Forsvar - {{ defender.name }}</option>

    {% endfor %}

    {% for midfielder in midfielders %}

    <option name="deleteplayer" value="m{{ midfielder.id }}">Midtbane - {{ midfielder.name }}</option>

    {% endfor %}

    {% for attacker in attackers%}

    <option name="deleteplayer" value="a{{ attacker.id }}">Angriber - {{ attacker.name }}</option>

    {% endfor %}

  </select>

  <input type="submit" value="Slet">

</form>

</body>

</html>

## Createattacker.html

<!doctype html>

<html>

<head>

<title>Opret spiller</title>

</head>

<body>

<h1>Opret Angriber</h1>

<form method="post" action="/createattacker">

<input type="text" name="name" id="name" placeholder="Navn">

<input type="number" name="age" id="age" placeholder="Alder">

<input type="text" name="team" id="team" placeholder="Hold">

<input type="number" name="goals" id="goals" placeholder="Mål">

<input type="submit" value="Opret">

</form>

</body>

</html>

## Createdefender.html

<!doctype html>

<html>

<head>

<title>Opret spiller</title>

</head>

<body>

<h1>Opret forsvar</h1>

<form method="post" action="/createdefender">

<input type="text" name="name" id="name" placeholder="Navn">

<input type="number" name="age" id="age" placeholder="Alder">

<input type="text" name="team" id="team" placeholder="Hold">

<input type="number" name="blocked" id="blocked" placeholder="Bolde blokeret">

<input type="submit" value="Opret">

</form>

</body>

</html>

## Createmidfielder.html

<!doctype html>

<html>

<head>

<title>Opret spiller</title>

</head>

<body>

<h1>Opret Midtbane</h1>

<form method="post" action="/createmidfielder">

<input type="text" name="name" id="name" placeholder="Navn">

<input type="number" name="age" id="age" placeholder="Alder">

<input type="text" name="team" id="team" placeholder="Hold">

<input type="number" name="passes" id="passes" placeholder="Afleveringer">

<input type="submit" value="Opret">

</form>

</body>

</html>

<form method="post" action="/createmidfielder">

    <input type="submit" value="Opret">

</form>

## Creategoalkeeper.html

<!doctype html>

<html>

<head>

<title>Opret spiller</title>

</head>

<body>

<h1>Opret Målmand</h1>

<form method="post" action="/creategoalkeeper">

<input type="text" name="name" id="name" placeholder="Navn">

<input type="number" name="age" id="age" placeholder="Alder">

<input type="text" name="team" id="team" placeholder="Hold">

<input type="number" name="saves" id="saves" placeholder="Redninger">

<input type="submit" value="Opret">

</form>

</body>

</html>