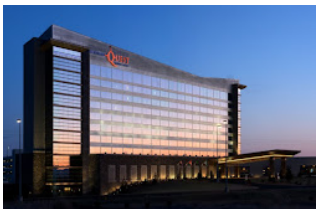
7/8/2022

Northern Quest Resort & Casino Machine Activity

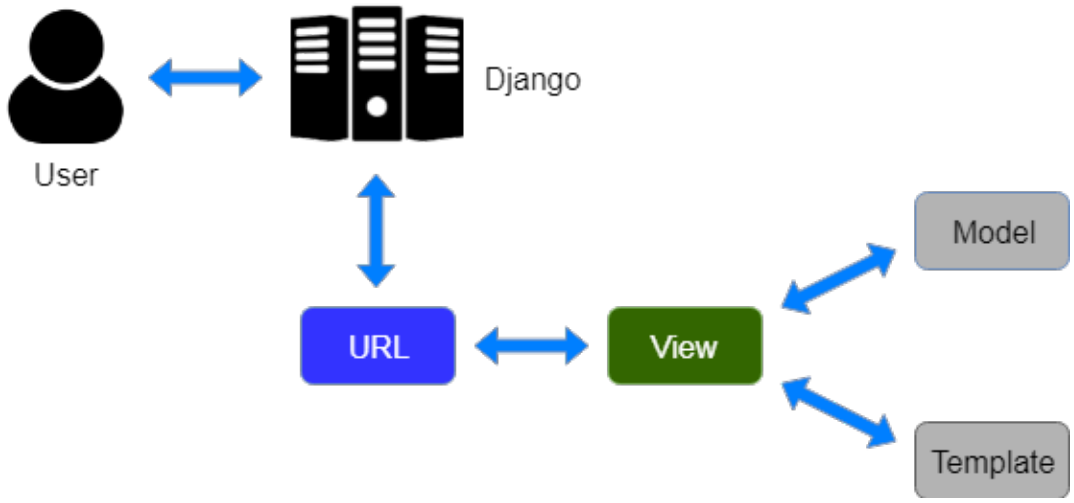
Service Layer Design

Keith Bruyer
MARYVILLE UNIVERSITY
SWDV 691 SOFTWARE DEVELOPMENT CAPSTONE



Overview

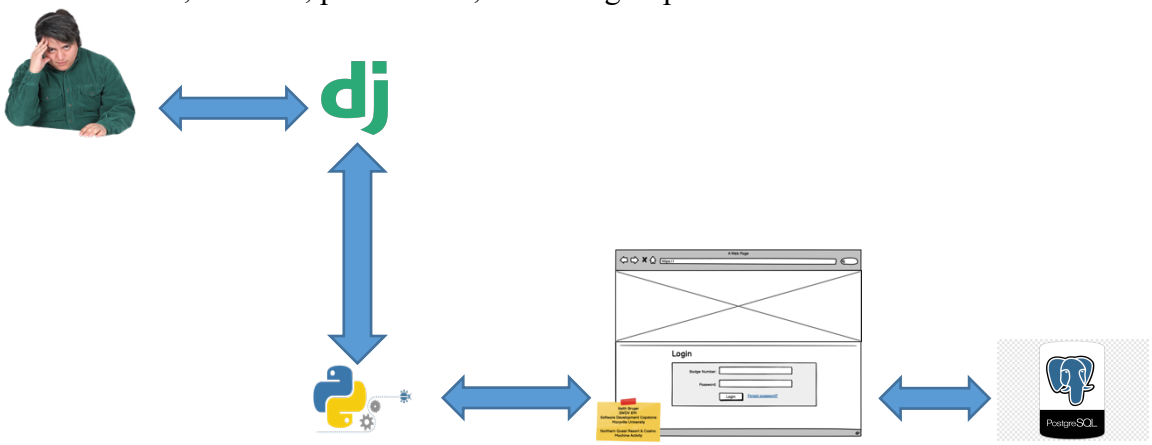
The backend service for this website will be Django, which uses request and response objects to pass states through the system. In this way, when one of the pages is requested, Django will create an HttpRequest object that will contain metadata about the request. Django will then load the appropriate view, passing the HttpRequest as the first argument to the view function. From there, each view will be returning an HttpResponse object. For example:



For each page in the application, I have provided how Django as the service layer will be developed.

I. Login:

Django comes with a built-in authentication framework that can handle user authentication, sessions, permissions, and user groups.



Login:
Method: POST
Middleware: AuthenticationMiddleware, SessionMiddleware
Purpose: When an employee logs in successfully, an HttpResponse object will be returned, which will trigger the rendering of the home page.
Example requests code:
accounts/url.py

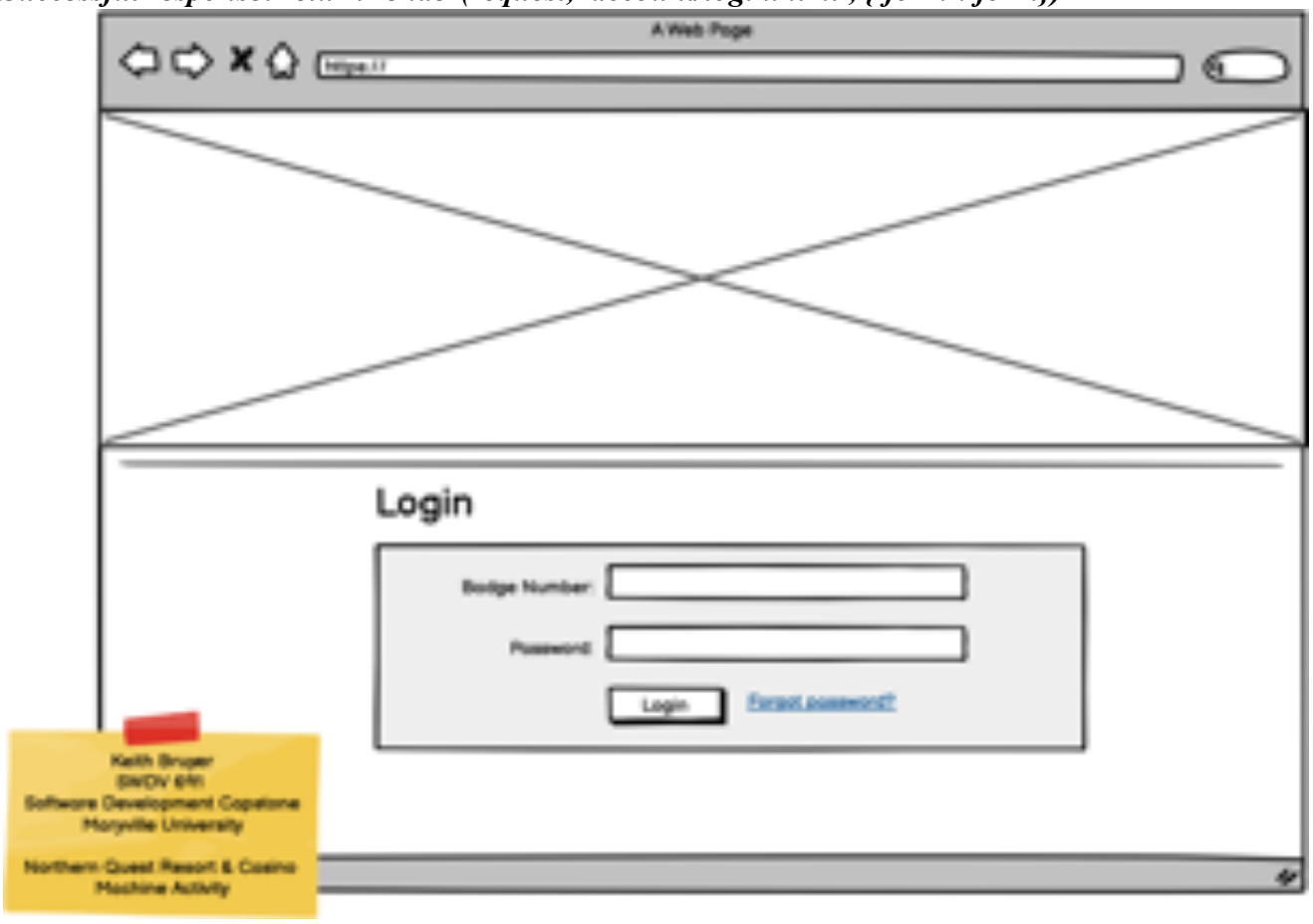
```
urlpatterns = [
  path('login/', auth_views.LoginView.as_view(), name='login'),
  path('logout/', auth_views.LogoutView.as_view(), name='logout')

  path('', views.dashboard, name='dashboard'), path('',
  include('django.contrib.auth.urls')), path('register/',
  views.register, name='register'), path('edit/', views.edit,
  name='edit'),
]
```

accounts/views.py

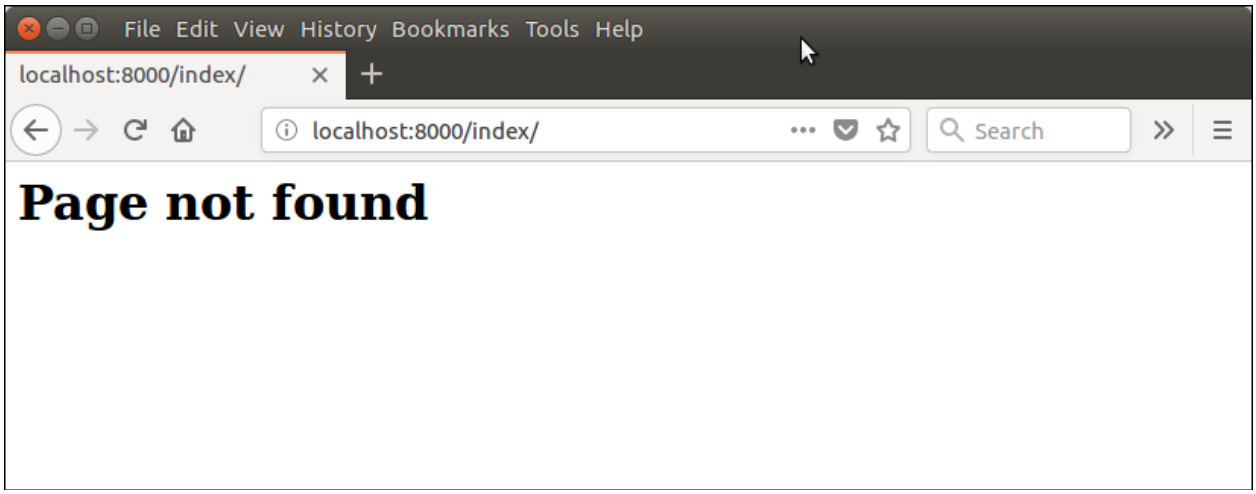
```
from django.http import HttpResponse
from django.shortcuts import render
from django.contrib.auth import authenticate, login
from django.contrib.auth.decorators import login_required
from django.contrib import messages
from .models import Profile
from .forms import LoginForm, UserRegistrationForm, \
                   UserEditForm, ProfileEditForm

def user_login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data
            user = authenticate(request,
                                username=cd['username'],
                                password=cd['password'])
```

```
            if user is not None:
                if user.is_active:
                    login(request, user)
                    return HttpResponse('Authenticated '\
                                        'successfully')
                else:
                    return HttpResponse('Disabled account')
            else:
                return HttpResponse('Invalid login')
    else:
        form = LoginForm()
    return render(request, 'account/login.html', {'form': form})
```

Successful response: return render(request, 'account/login.html', {'form': form})

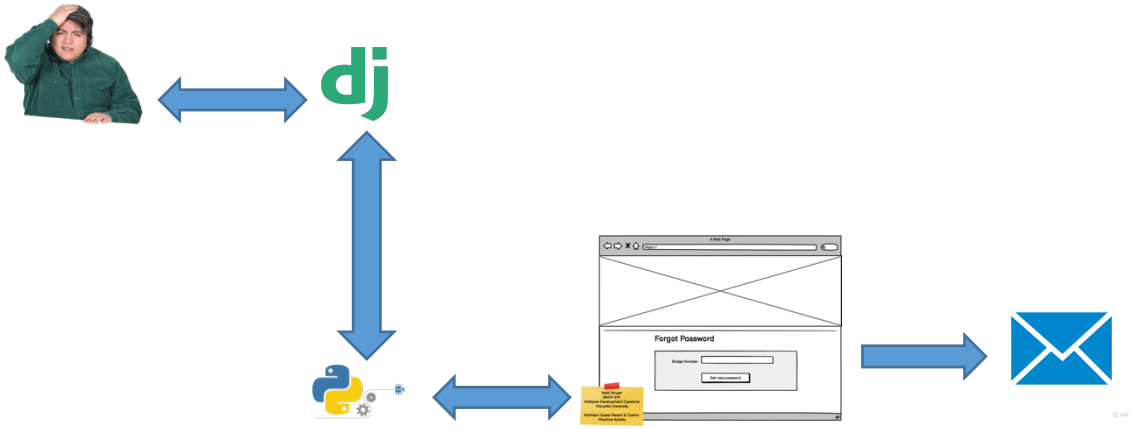


Unsuccessful response: return HttpResponseNotFound('<h1>Page not found</h1>')


File Edit View History Bookmarks Tools Help
localhost:8000/index/
localhost:8000/index/
Search
Page not found

II. Forgot Password:

Django comes with a built-in framework that can handle the creation and send of emails.



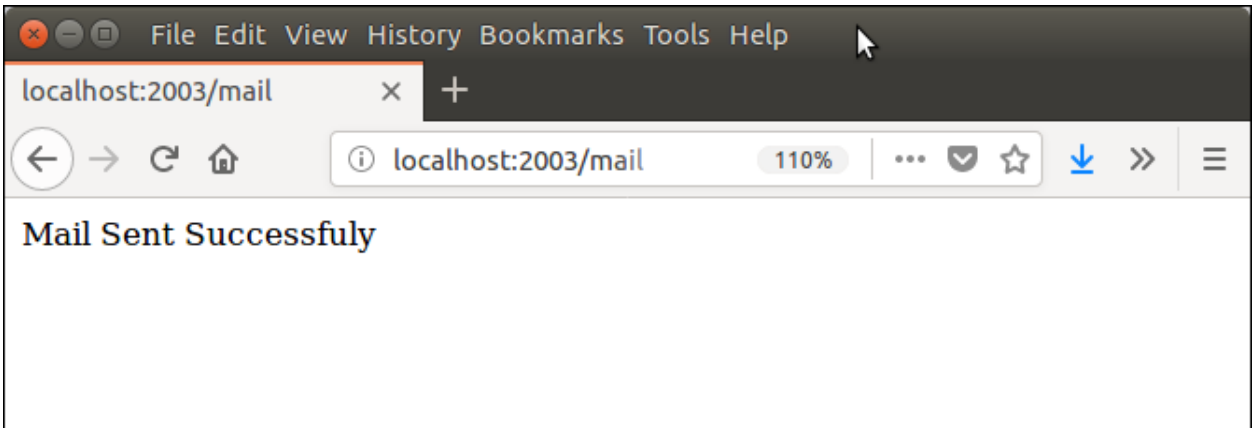
Forgot Password:
Method: POST
Middleware: SessionMiddleware
Purpose: When an employee clicks on the forgot password link, an HttpResponse object will be sent, which will trigger a password reset email to be sent.
Example requests code:
machine-activity/settings.py

```
EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
EMAIL_HOST = 'smtp.northernquest.com'
EMAIL_HOST_USER = '<admin account>@northernquest.com'
EMAIL_HOST_PASSWORD = '<to-be-determined>'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
```

Machine-activity/views.py

```
from django.http import HttpResponse
from djangpapp import settings
from django.core.mail import send_mail


def mail(request):
    subject = "Greetings"
    msg     = "Congratulations for your success"
    to      = "irfan.sssit@gmail.com"
    res     = send_mail(subject, msg, settings.EMAIL_HOST_USER, [to])
    if(res == 1):
        msg = "Mail Sent Successfuly"
    else:
        msg = "Mail could not sent"
    return HttpResponse(msg)
```

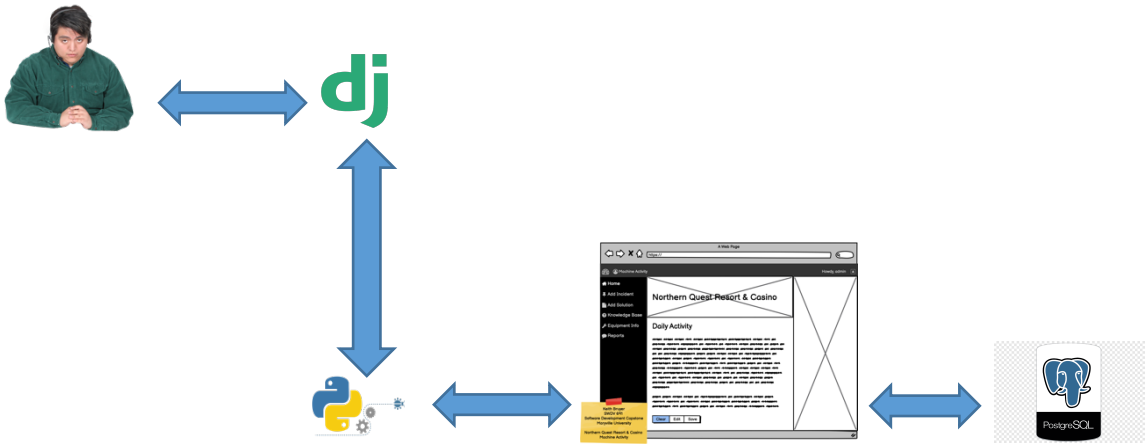
Successful response:



Unsuccessful response: return HttpResponse(*"Mail could not sent"****)***

III. Home:

This app will use Django views which are Python functions that take HTTP requests and return an HTTP response, like HTML documents. Views for this app will be put in a file called views.py which will be in the app's folder.

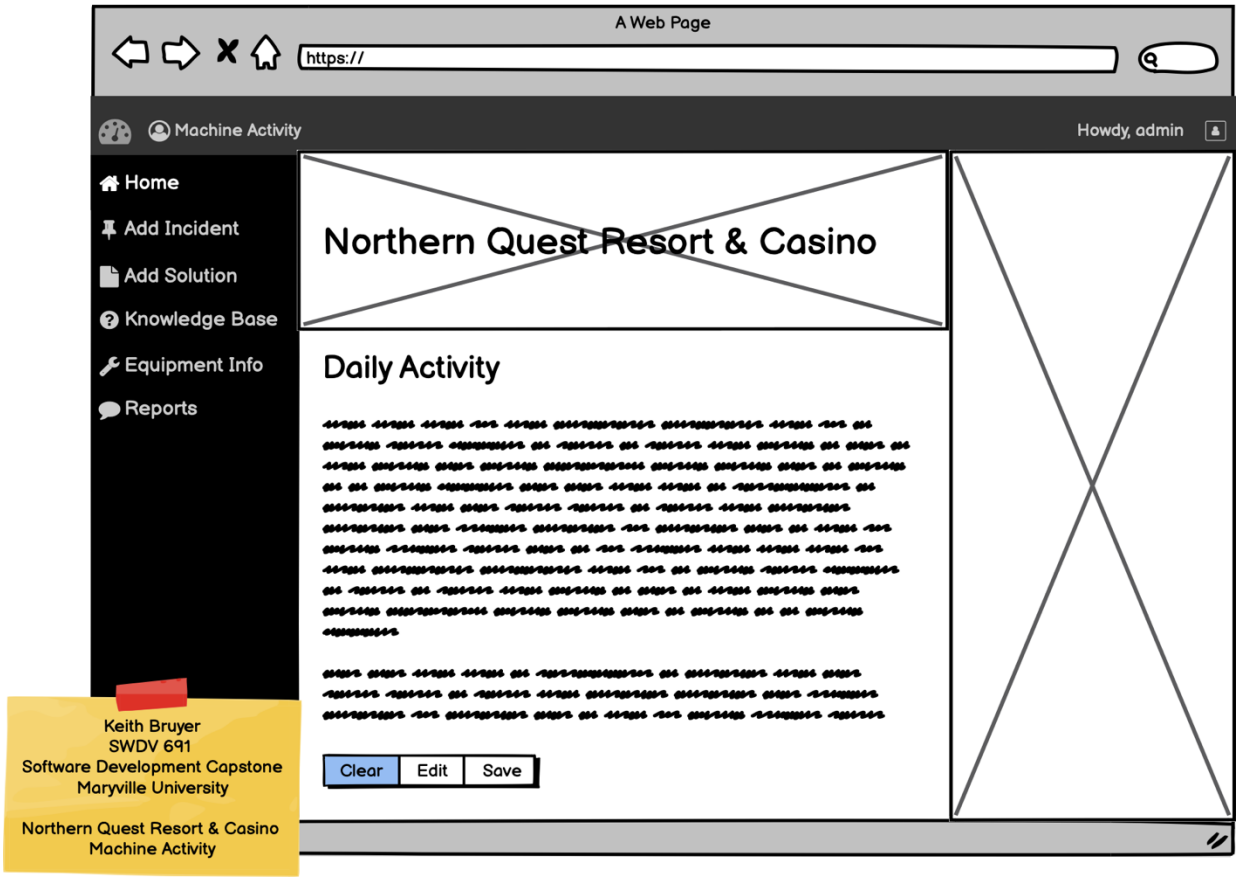


Home:

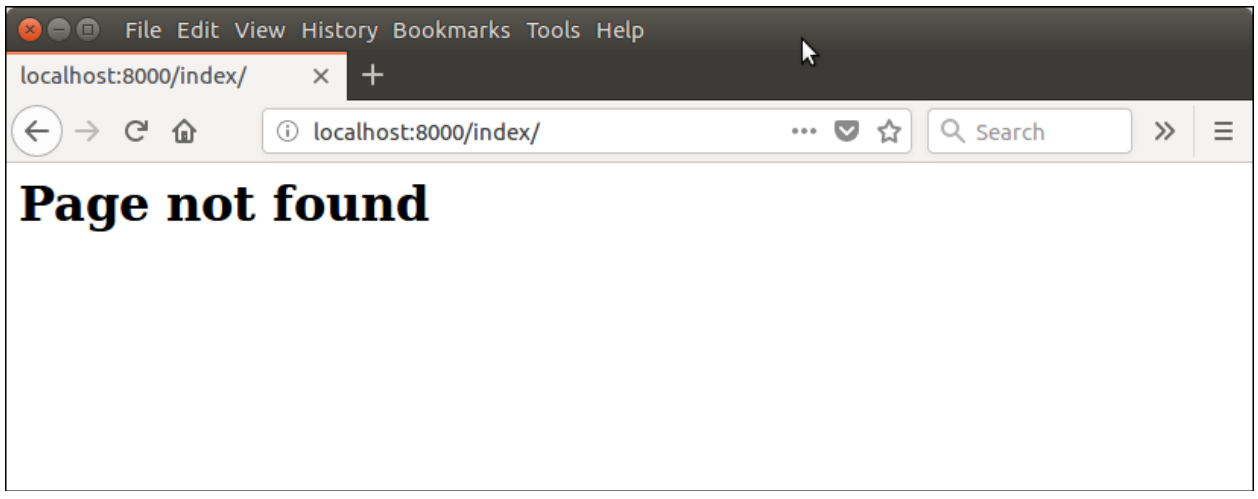
Method: GET
URL: https://machine-activity.herokuapp.com/index.html
Purpose: When an employee logs in successfully, this endpoint will be called from the front end and the home page returned and rendered in the browser.
Example requests:

```
// urls.py
path('index/', views.index),

// views.py
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseNotFound

def index(request):
    a = 1
    if a:
        return HttpResponseNotFound('<h1>Page not found</h1>')
    else:
        return HttpResponse('home page') # rendering the template in HttpResponse
```

Successful response: return HttpResponse('home page')

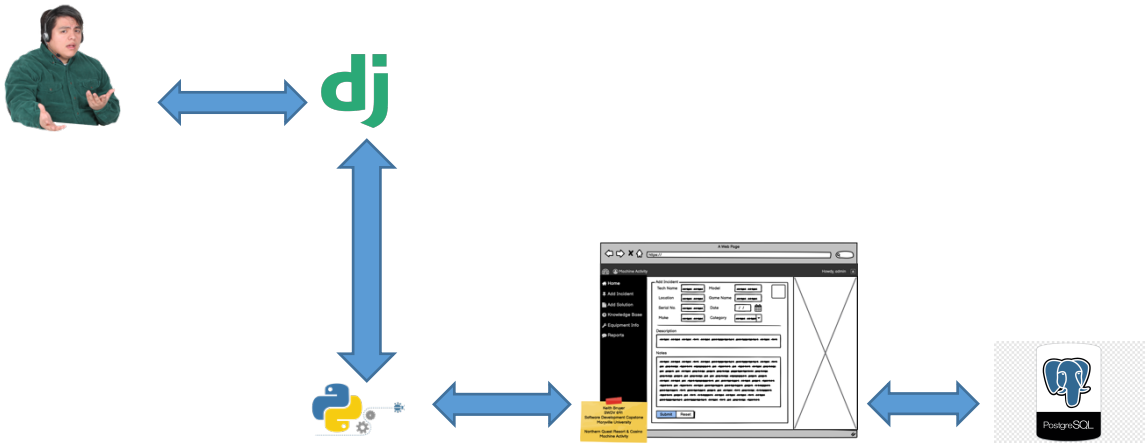


Unsuccessful response: return HttpResponseNotFound('<h1>Page not found</h1>')



IV. Add Incident:

This app will use Django views which are Python functions that take HTTP requests and return an HTTP response, like HTML documents. Views for this app will be put in a file called views.py which will be in the app's folder.



Add Incident (page):

Method: GET

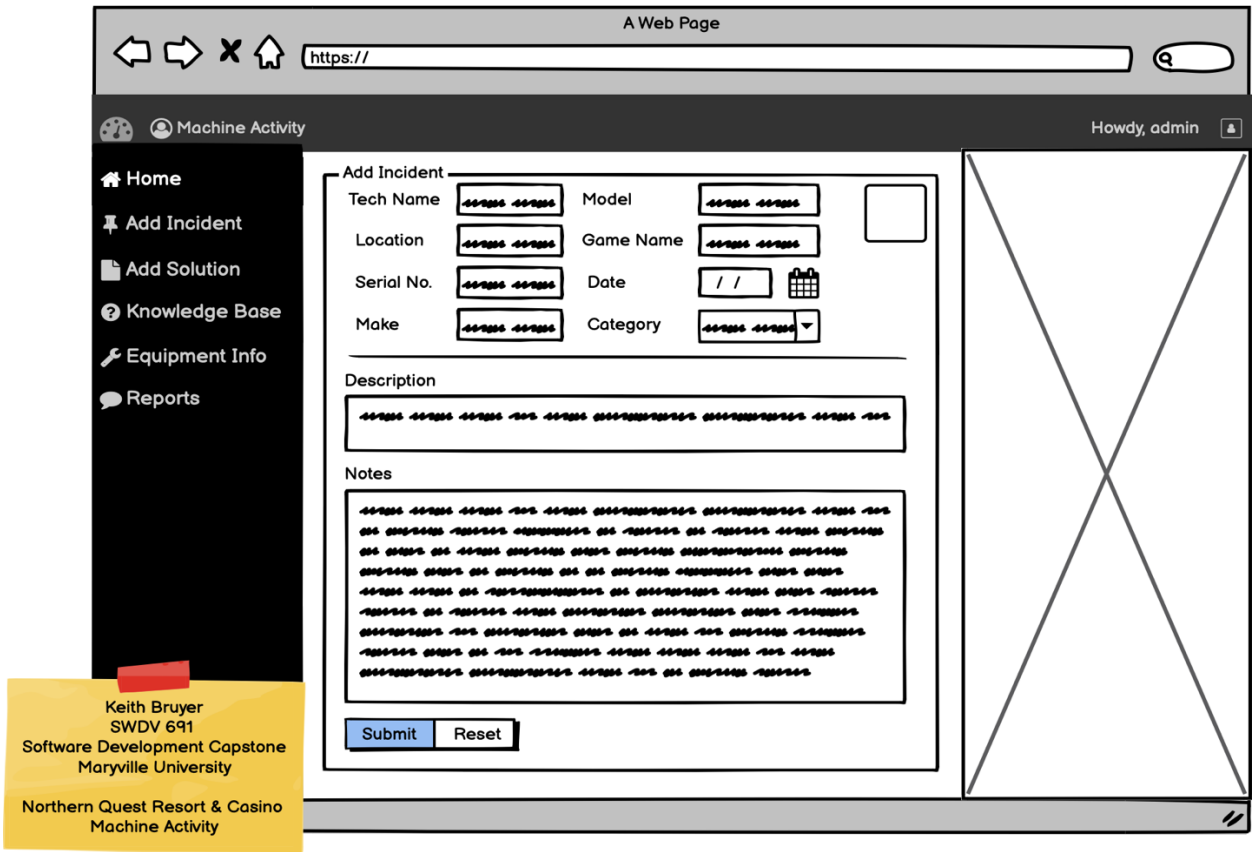
URL: https://machine-activity.herokuapp.com/incident.html

Purpose: This endpoint is called from the navigation bar and allows the user to input and post an incident to the database.

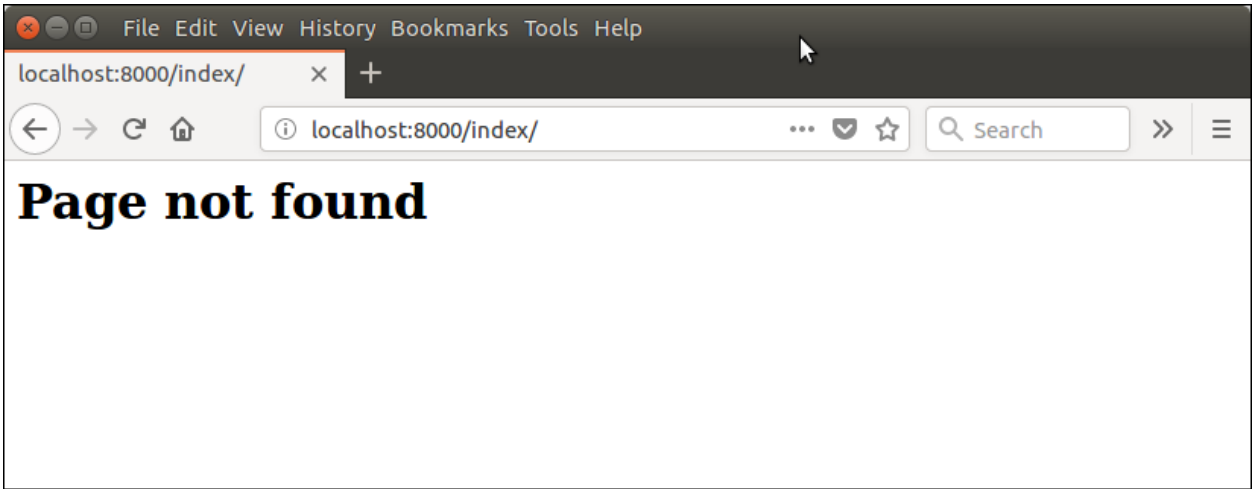
Example requests:

```
// urls.py
path('incident/', views.incident),

// views.py
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseNotFound

def index(request):
    a = 1
    if a:
        return HttpResponseNotFound('<h1>Page not found</h1>')
    else:
        return HttpResponse('incident page') # rendering the template in
HttpResponse
```

Successful response: return HttpResponse('incident page')



Unsuccessful response: return HttpResponseNotFound('<h1>Page not found</h1>')



Add Incident (action):

Method: POST

URL: https://machine-activity.herokuapp.com/incident.html

Purpose: Upon entering the information into the form and clicking on the submit button, the data will be posted to the database.

Example requests:

```
from django.db import models
from django.urls import reverse


class Post(models.Model):
    techName = models.TextField(max_length=50)
    location = models.TextField(max_length=50)
    body = models.TextField()
    serialNumber = models.IntField()
    make = models.TextField(max_length=50)
    model = models.TextField(max_length=50)
    gameName = models.TextField(max_length=50)
    date = Date()
    category = (
       TABLE_GAMES = 'TG'
       CAGE = 'CG'
       RECYCLER = 'RC'
       OTB = 'OT'
       BENCH = 'BN'
       VENDOR = 'VN'
       NRT = 'NR'
       SLOT_MACHINES = 'SM'
       MISC = 'MI'
       CATEGORIES = [
           (TABLE_GAMES, 'Table Games'),
           (CAGE, 'Cage'),
           (RECYCLER, 'Recycler'),
           (OTB, 'OTB'),
           (BENCH, 'Bench'),
           (VENDOR, 'Vendor'),
           (NRT, 'NRT'),
           (SLOT_MACHINES, 'Slot Machines'),
           (MISC, 'Misc')
       ]
       category = models.CharField(
           max_length=2,
           choices=CATEGORIES,
           default=SLOT_MACHINES,
       )

    def get_absolute_url(self):
        return reverse("incident page", kwargs={"pk": self.pk})
```

Successful response: The form will clear for another entry and an alert will pop up.

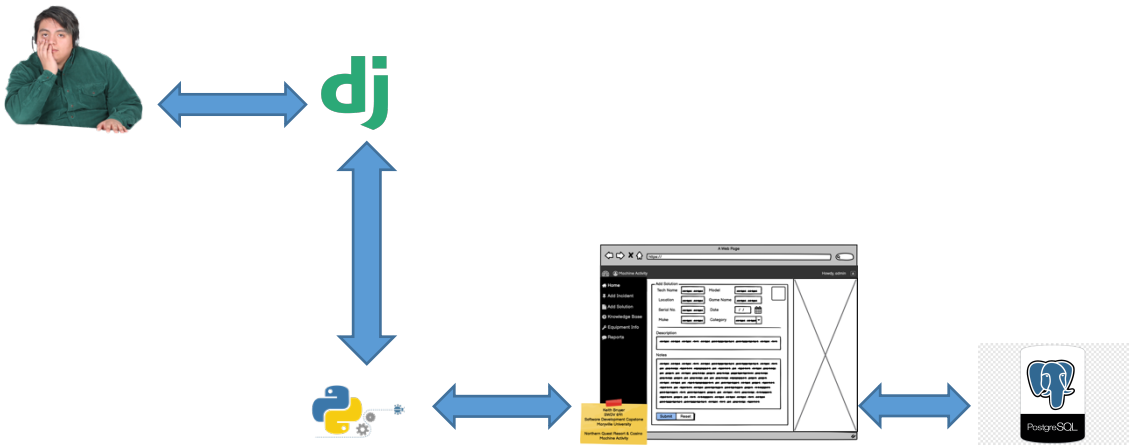
Unsuccessful response: One of the following exceptions will be thrown depending on the error.

Exception	Description
DatabaseError	It occurs when the database is not available.

IntegrityError	It occurs when an insertion query executes.
DataError	It raises when data-related issues come into the database.

V. Add Solution:

This app will use Django views which are Python functions that take HTTP requests and return an HTTP response, like HTML documents. Views for this app will be put in a file called views.py which will be in the app's folder.



Add Solution (page):

Method: GET

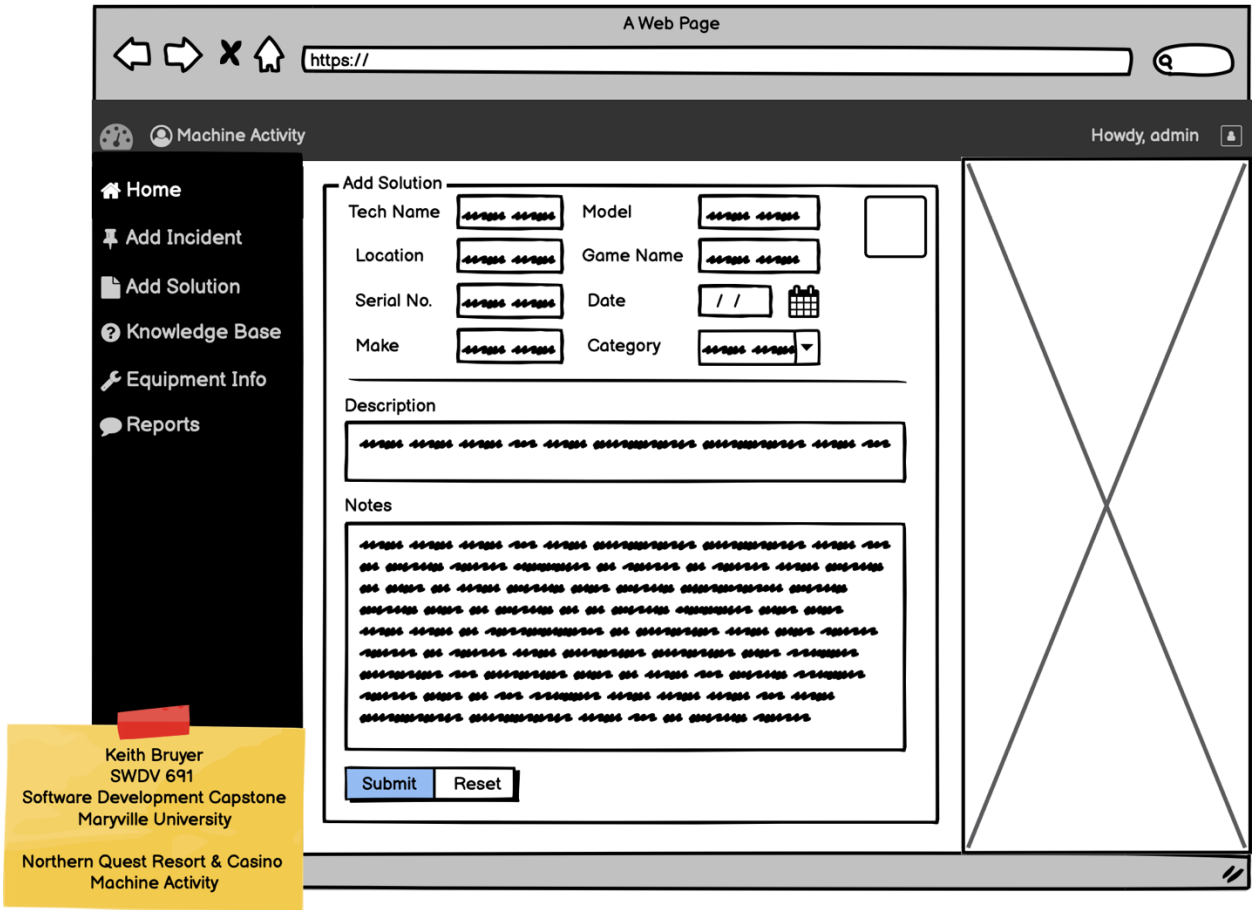
URL: https://machine-activity.herokuapp.com/incident.html

Purpose: This endpoint is called from the navigation bar and allows the user to input and post a solution to the database.

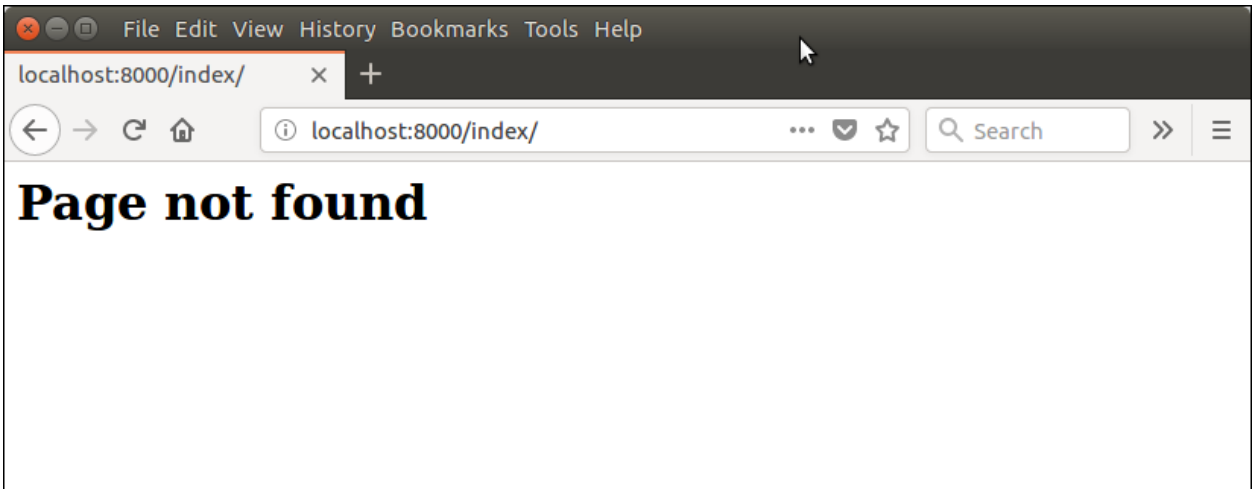
Example requests:

```
// urls.py
path('solution/', views.solution),

// views.py
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseNotFound

def index(request):
    a = 1
    if a:
        return HttpResponseNotFound('<h1>Page not found</h1>')
    else:
        return HttpResponse('incident page') # rendering the template in
HttpResponse
```

Successful response: return HttpResponse('solution page')



Unsuccessful response: return HttpResponseNotFound('<h1>Page not found</h1>')



Add solution (action):

Method: POST

URL: https://machine-activity.herokuapp.com/solution.html

Purpose: Upon entering the information into the form and clicking on the submit button, the data will be posted to the database.

Example requests:

```
from django.db import models
from django.urls import reverse


class Post(models.Model):
    techName = models.TextField(max_length=50)
    location = models.TextField(max_length=50)
    body = models.TextField()
    serialNumber = models.IntField()
    make = models.TextField(max_length=50)
    model = models.TextField(max_length=50)
    gameName = models.TextField(max_length=50)
    date = Date()
    category = (
       TABLE_GAMES = 'TG'
       CAGE = 'CG'
       RECYCLER = 'RC'
       OTB = 'OT'
       BENCH = 'BN'
       VENDOR = 'VN'
       NRT = 'NR'
       SLOT_MACHINES = 'SM'
       MISC = 'MI'
       CATEGORIES = [
           (TABLE_GAMES, 'Table Games'),
           (CAGE, 'Cage'),
           (RECYCLER, 'Recycler'),
           (OTB, 'OTB'),
           (BENCH, 'Bench'),
           (VENDOR, 'Vendor'),
           (NRT, 'NRT'),
           (SLOT_MACHINES, 'Slot Machines'),
           (MISC, 'Misc')
       ]
       category = models.CharField(
           max_length=2,
           choices=CATEGORIES,
           default=SLOT_MACHINES,
       )

    def get_absolute_url(self):
        return reverse("incident page", kwargs={"pk": self.pk})
```

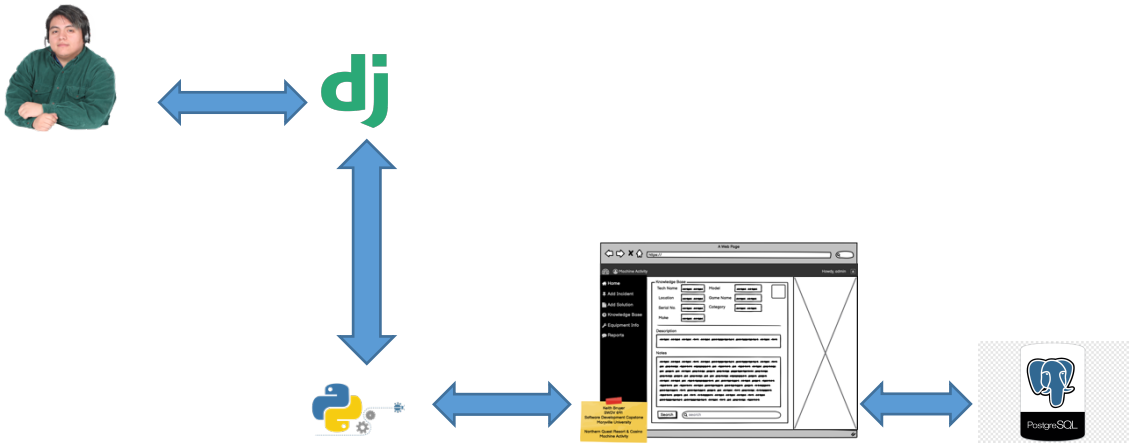
Successful response: The form will clear for another entry and an alert will pop up.

Unsuccessful response: One of the following exceptions will be thrown depending on the error.

Exception	Description
DatabaseError	It occurs when the database is not available.
IntegrityError	It occurs when an insertion query executes.
DataError	It raises when data-related issues come into the database.

VI. Knowledge Base:

This app will use Django views which are Python functions that take HTTP requests and return an HTTP response, like HTML documents. Views for this app will be put in a file called views.py which will be in the app's folder.



Knowledge Base (page):

Method: GET

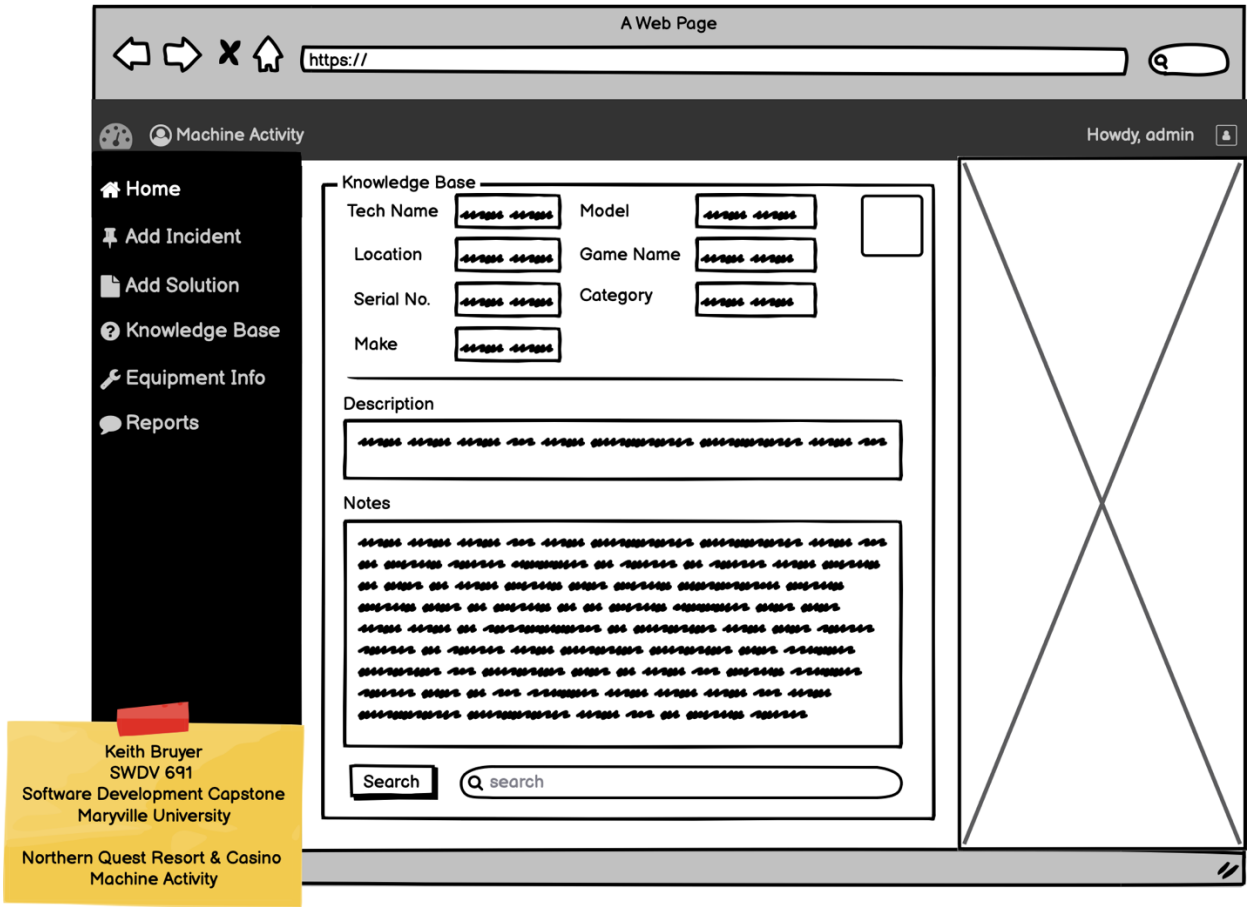
URL: https://machine-activity.herokuapp.com/knowledgeBase.html

Purpose: This endpoint is called from the navigation bar and allows the user to input and get a solution from the database.

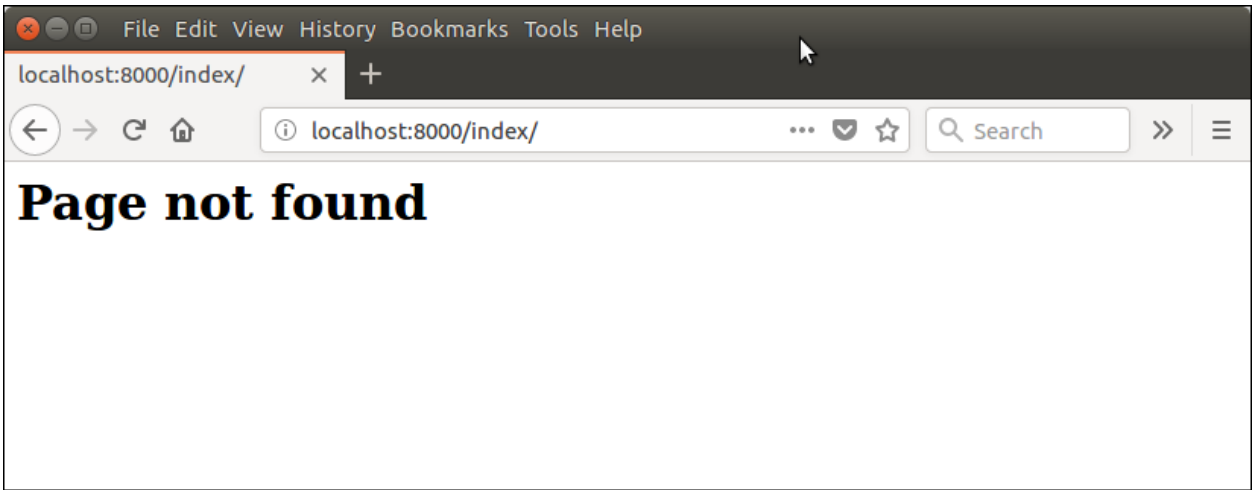
Example requests:

```
// urls.py
path('solution/', views.solution),

// views.py
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseNotFound

def index(request):
    a = 1
    if a:
        return HttpResponseNotFound('<h1>Page not found</h1>')
    else:
        return HttpResponse('incident page') # rendering the template in
HttpResponse
```

Successful response: return HttpResponse(knowledgeBase page')



Unsuccessful response: return HttpResponseNotFound('<h1>Page not found</h1>')



Get solution (action):

Method: GET
URL: https://machine-activity.herokuapp.com/knowledgeBase.html
Purpose: This endpoint is called from the navigation bar and allows the user to input and get a solution from the database.
Example requests:

```
from django import forms

class SearchForm(forms.Form):
    query = forms.CharField()
```

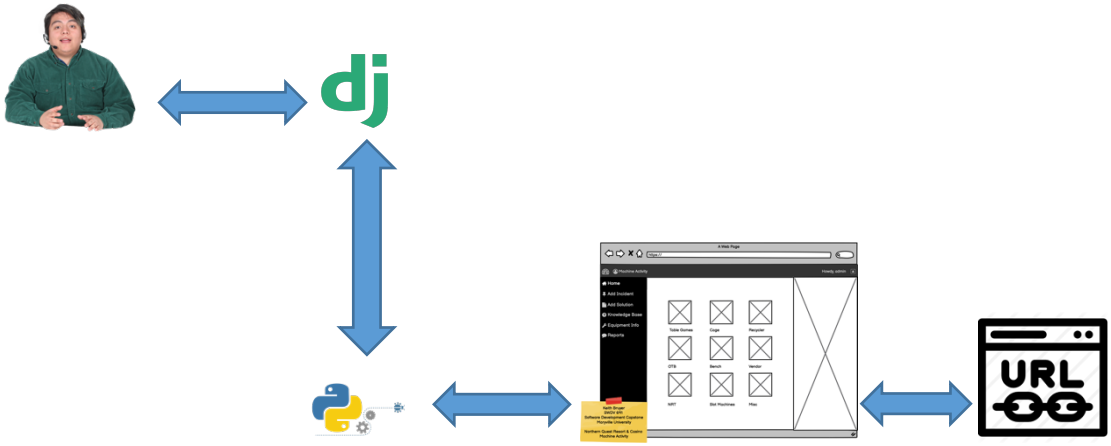
Successful response: The query result will render on a new page.

Unsuccessful response: One of the following exceptions will be thrown depending on the error.

Exception	Description
DatabaseError	It occurs when the database is not available.
IntegrityError	It occurs when an insertion query executes.
DataError	It raises when data-related issues come into the database.

VII. Equipment Info:

This app will use Django views which are Python functions that take HTTP requests and return an HTTP response, like HTML documents. Views for this app will be put in a file called views.py which will be in the app's folder.



Get Equipment Info:

Method: GET

URL: https://machine-activity.herokuapp.com/equipmentInfo.html

Purpose: This endpoint is called from the navigation bar and allows the user to get equipment information from vendor sites.

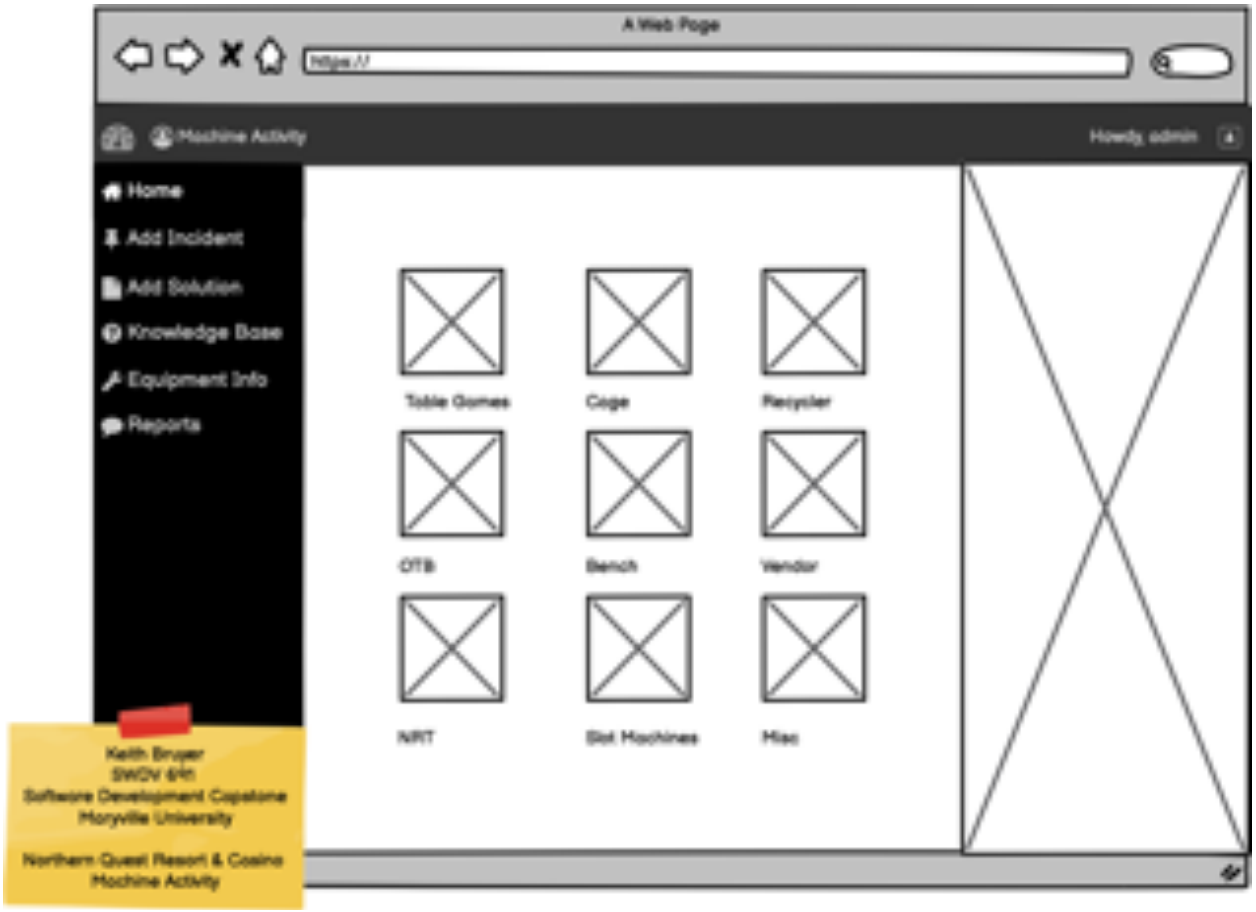
Example requests:

```
// urls.py
path('solution/', views.pageInfo),

// views.py
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseNotFound

def index(request):
    a = 1
    if a:
        return HttpResponseNotFound('<h1>Page not found</h1>')
    else:
        return HttpResponse('incident page') # rendering the template in
HttpResponse
```

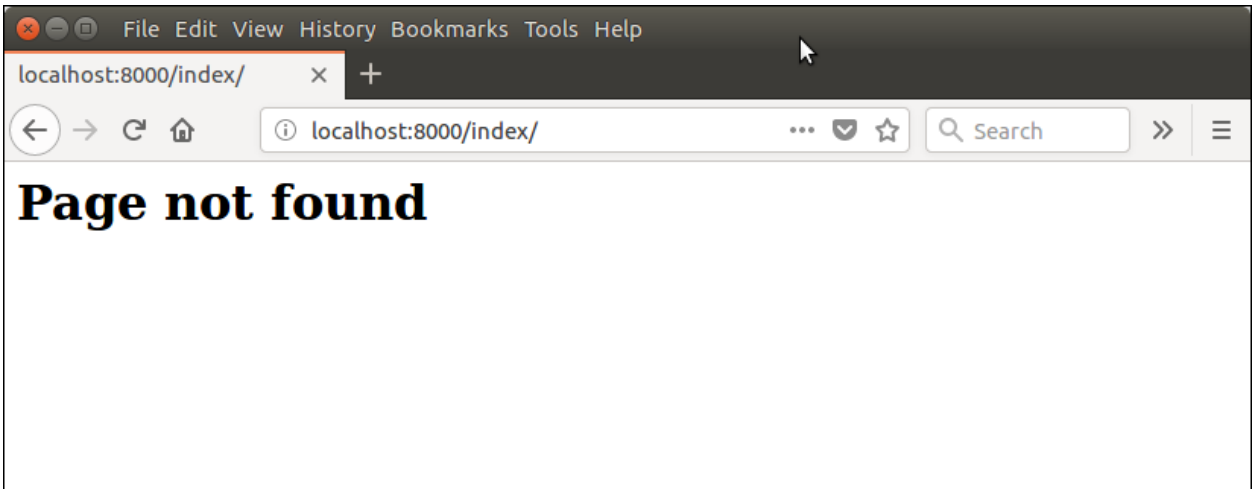
***** Each image on the screen will have an embedded link such as the following:***

```
<a href="https://www.<vendorsite>/">Visit Vendor Site</a>
```

Successful response: return HttpResponse(equipmentInfo.html')

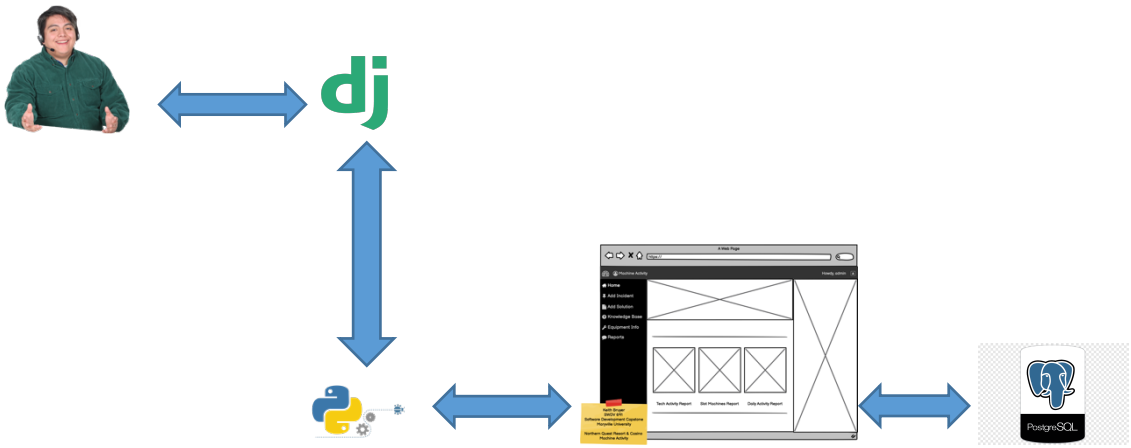


Unsuccessful response: return HttpResponseNotFound('<h1>Page not found</h1>')



VIII. Reports:

This app will use Django views which are Python functions that take HTTP requests and return an HTTP response, like HTML documents. Views for this app will be put in a file called views.py which will be in the app's folder.

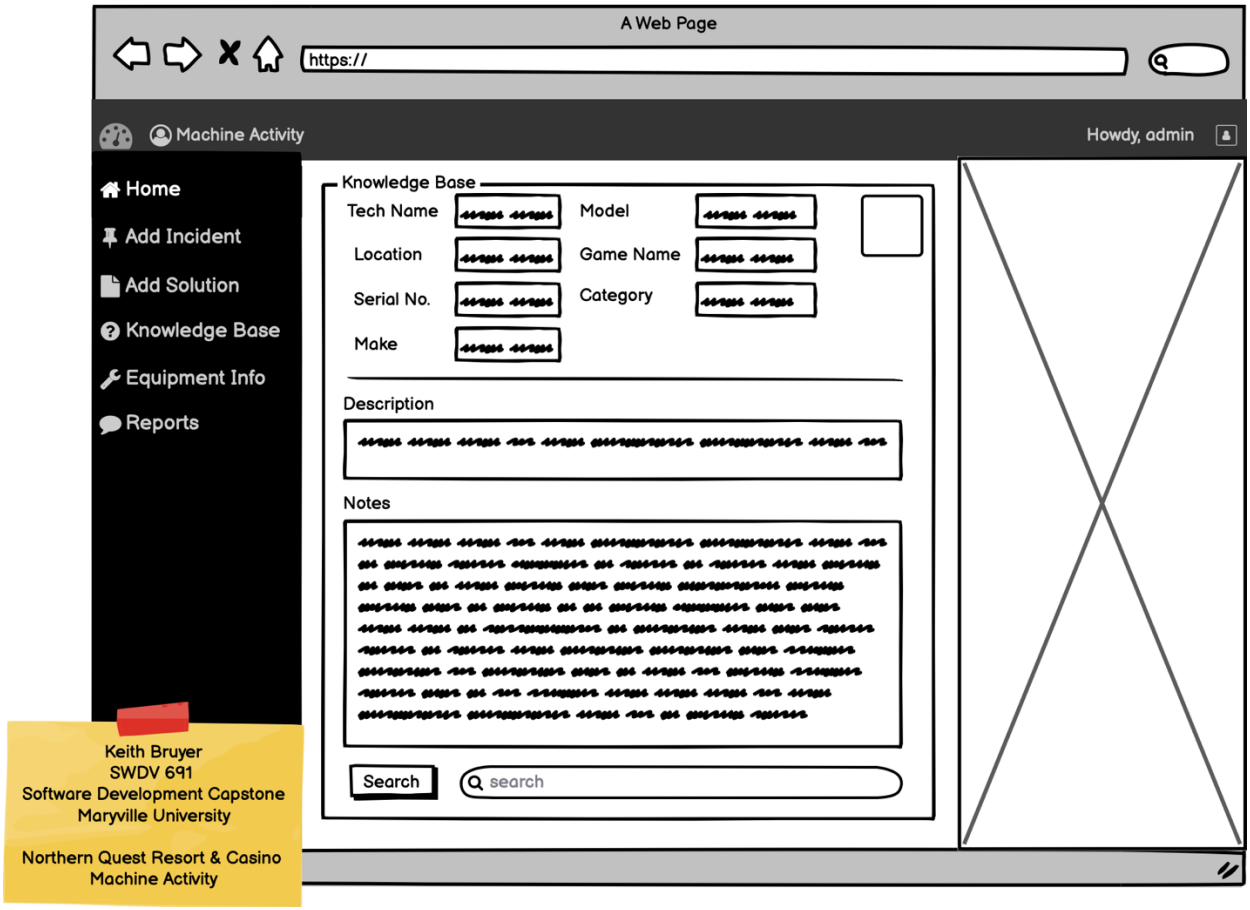


Get Report:

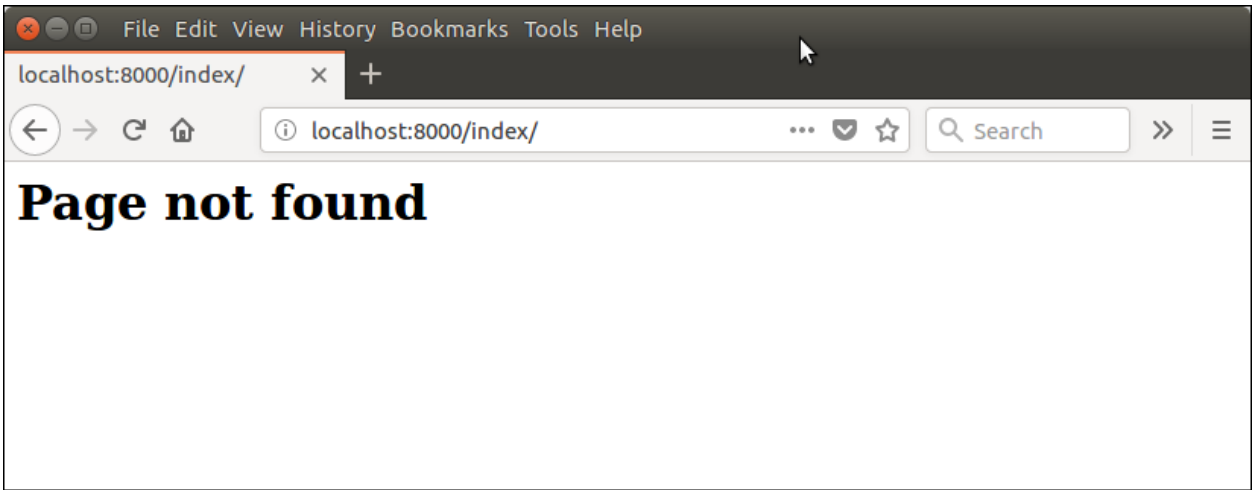
Method: GET
URL: https://machine-activity.herokuapp.com/reports.html
Purpose: This endpoint is called from the navigation bar and allows the user to get predefined reports queried from the database.
Example requests:

```
// urls.py
path('solution/', views.reports),

// views.py
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseNotFound

def index(request):
    a = 1
    if a:
        return HttpResponseNotFound('<h1>Page not found</h1>')
    else:
        return HttpResponse('incident page') # rendering the template in
HttpResponse
```

Successful response: return HttpResponse(reports.page')



Unsuccessful response: return HttpResponseNotFound('<h1>Page not found</h1>')



Get report (action):

Method: GET

URL: https://machine-activity.herokuapp.com/reports.html

Purpose: This endpoint is called from the navigation bar and allows the user to get predefined reports queried from the database.

Example requests:

```
from django import forms

class SearchForm(forms.Form):
   reportOneQuery = forms.CharField()
   reportTwoQuery = forms.CharField()
   reportThreeQuery = forms.CharField()
```

Successful response: The query result will render on a new page.

Unsuccessful response: One of the following exceptions will be thrown depending on the error.

Exception	Description
DatabaseError	It occurs when the database is not available.
IntegrityError	It occurs when an insertion query executes.
DataError	It raises when data-related issues come into the database.