In [1]:
```python
# nadam_optimizer.py

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import cifar10
from tensorflow.keras import layers, models
from tensorflow.keras.optimizers import Nadam
from sklearn.metrics import confusion_matrix
import seaborn as sns
import time
from sklearn.model_selection import train_test_split
```

In [2]:
```python
# Load CIFAR-10 dataset
(X_train, y_train), (X_test, y_test) = cifar10.load_data()

# Normalize the images
X_train = X_train.astype('float32') / 255.0
X_test = X_test.astype('float32') / 255.0

# Ensure labels are integers (no one-hot encoding)
y_train = y_train.astype('int')
y_test = y_test.astype('int')

X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2,
print (X_train.shape )
print (X_test.shape )
```

```
(40000, 32, 32, 3)
(10000, 32, 32, 3)
```

In [3]:
```python
# Create model function
def create_model(optimizer):
    model = models.Sequential()
    model.add(layers.Flatten(input_shape=(32, 32, 3)))
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(10, activation='softmax'))
    model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy', metr
    return model

# Nadam optimizer
optimizer = Nadam(learning_rate=0.001)
```

In [4]:
```python
# Train and evaluate model
start_time = time.time()
model = create_model(optimizer)
history = model.fit(X_train, y_train, epochs=50, batch_size=64, validation_data=(X_
end_time = time.time()

# Record training time
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")
```

```
Epoch 1/50
625/625 [==============================] - 10s 14ms/step - loss: 1.9437 - accuracy:
0.3023 - val_loss: 1.7984 - val_accuracy: 0.3550
Epoch 2/50
625/625 [==============================] - 5s 8ms/step - loss: 1.7543 - accuracy: 0.
3741 - val_loss: 1.7134 - val_accuracy: 0.3882
Epoch 3/50
625/625 [==============================] - 5s 8ms/step - loss: 1.6829 - accuracy: 0.
4013 - val_loss: 1.6866 - val_accuracy: 0.3963
Epoch 4/50
625/625 [==============================] - 6s 10ms/step - loss: 1.6408 - accuracy:
0.4199 - val_loss: 1.6530 - val_accuracy: 0.4027
Epoch 5/50
625/625 [==============================] - 5s 7ms/step - loss: 1.6085 - accuracy: 0.
4316 - val_loss: 1.6087 - val_accuracy: 0.4271
Epoch 6/50
625/625 [==============================] - 5s 9ms/step - loss: 1.5837 - accuracy: 0.
4371 - val_loss: 1.5868 - val_accuracy: 0.4314
Epoch 7/50
625/625 [==============================] - 6s 10ms/step - loss: 1.5628 - accuracy:
0.4443 - val_loss: 1.5824 - val_accuracy: 0.4381
Epoch 8/50
625/625 [==============================] - 5s 8ms/step - loss: 1.5438 - accuracy: 0.
4538 - val_loss: 1.5733 - val_accuracy: 0.4392
Epoch 9/50
625/625 [==============================] - 6s 9ms/step - loss: 1.5277 - accuracy: 0.
4586 - val_loss: 1.5753 - val_accuracy: 0.4313
Epoch 10/50
625/625 [==============================] - 6s 10ms/step - loss: 1.5154 - accuracy:
0.4645 - val_loss: 1.5955 - val_accuracy: 0.4285
Epoch 11/50
625/625 [==============================] - 5s 7ms/step - loss: 1.5040 - accuracy: 0.
4682 - val_loss: 1.5680 - val_accuracy: 0.4430
Epoch 12/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4911 - accuracy: 0.
4728 - val_loss: 1.5572 - val_accuracy: 0.4448
Epoch 13/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4806 - accuracy: 0.
4754 - val_loss: 1.5581 - val_accuracy: 0.4472
Epoch 14/50
625/625 [==============================] - 5s 7ms/step - loss: 1.4748 - accuracy: 0.
4760 - val_loss: 1.5483 - val_accuracy: 0.4470
Epoch 15/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4660 - accuracy: 0.
4803 - val_loss: 1.5528 - val_accuracy: 0.4414
Epoch 16/50
625/625 [==============================] - 6s 9ms/step - loss: 1.4585 - accuracy: 0.
4826 - val_loss: 1.5408 - val_accuracy: 0.4497
Epoch 17/50
625/625 [==============================] - 6s 9ms/step - loss: 1.4525 - accuracy: 0.
4858 - val_loss: 1.5542 - val_accuracy: 0.4443
Epoch 18/50
625/625 [==============================] - 5s 7ms/step - loss: 1.4450 - accuracy: 0.
4891 - val_loss: 1.5228 - val_accuracy: 0.4623
Epoch 19/50
625/625 [==============================] - 6s 10ms/step - loss: 1.4375 - accuracy:
```

```
0.4888 - val_loss: 1.5412 - val_accuracy: 0.4488
Epoch 20/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4314 - accuracy: 0.
4906 - val_loss: 1.5159 - val_accuracy: 0.4640
Epoch 21/50
625/625 [==============================] - 6s 10ms/step - loss: 1.4274 - accuracy:
0.4916 - val_loss: 1.5665 - val_accuracy: 0.4403
Epoch 22/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4216 - accuracy: 0.
4966 - val_loss: 1.5686 - val_accuracy: 0.4442
Epoch 23/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4162 - accuracy: 0.
4972 - val_loss: 1.5219 - val_accuracy: 0.4615
Epoch 24/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4129 - accuracy: 0.
4981 - val_loss: 1.5311 - val_accuracy: 0.4584
Epoch 25/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4093 - accuracy: 0.
5005 - val_loss: 1.5188 - val_accuracy: 0.4580
Epoch 26/50
625/625 [==============================] - 5s 8ms/step - loss: 1.4024 - accuracy: 0.
5011 - val_loss: 1.5210 - val_accuracy: 0.4626
Epoch 27/50
625/625 [==============================] - 5s 7ms/step - loss: 1.4003 - accuracy: 0.
5033 - val_loss: 1.5292 - val_accuracy: 0.4576
Epoch 28/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3965 - accuracy: 0.
5039 - val_loss: 1.5214 - val_accuracy: 0.4613
Epoch 29/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3920 - accuracy: 0.
5049 - val_loss: 1.5113 - val_accuracy: 0.4679
Epoch 30/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3854 - accuracy: 0.
5052 - val_loss: 1.5601 - val_accuracy: 0.4478
Epoch 31/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3829 - accuracy: 0.
5102 - val_loss: 1.5470 - val_accuracy: 0.4506
Epoch 32/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3804 - accuracy: 0.
5096 - val_loss: 1.5500 - val_accuracy: 0.4576
Epoch 33/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3764 - accuracy: 0.
5100 - val_loss: 1.5591 - val_accuracy: 0.4564
Epoch 34/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3736 - accuracy: 0.
5105 - val_loss: 1.5302 - val_accuracy: 0.4510
Epoch 35/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3688 - accuracy: 0.
5145 - val_loss: 1.5206 - val_accuracy: 0.4609
Epoch 36/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3674 - accuracy: 0.
5138 - val_loss: 1.5122 - val_accuracy: 0.4653
Epoch 37/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3648 - accuracy: 0.
5136 - val_loss: 1.5315 - val_accuracy: 0.4583
Epoch 38/50
```

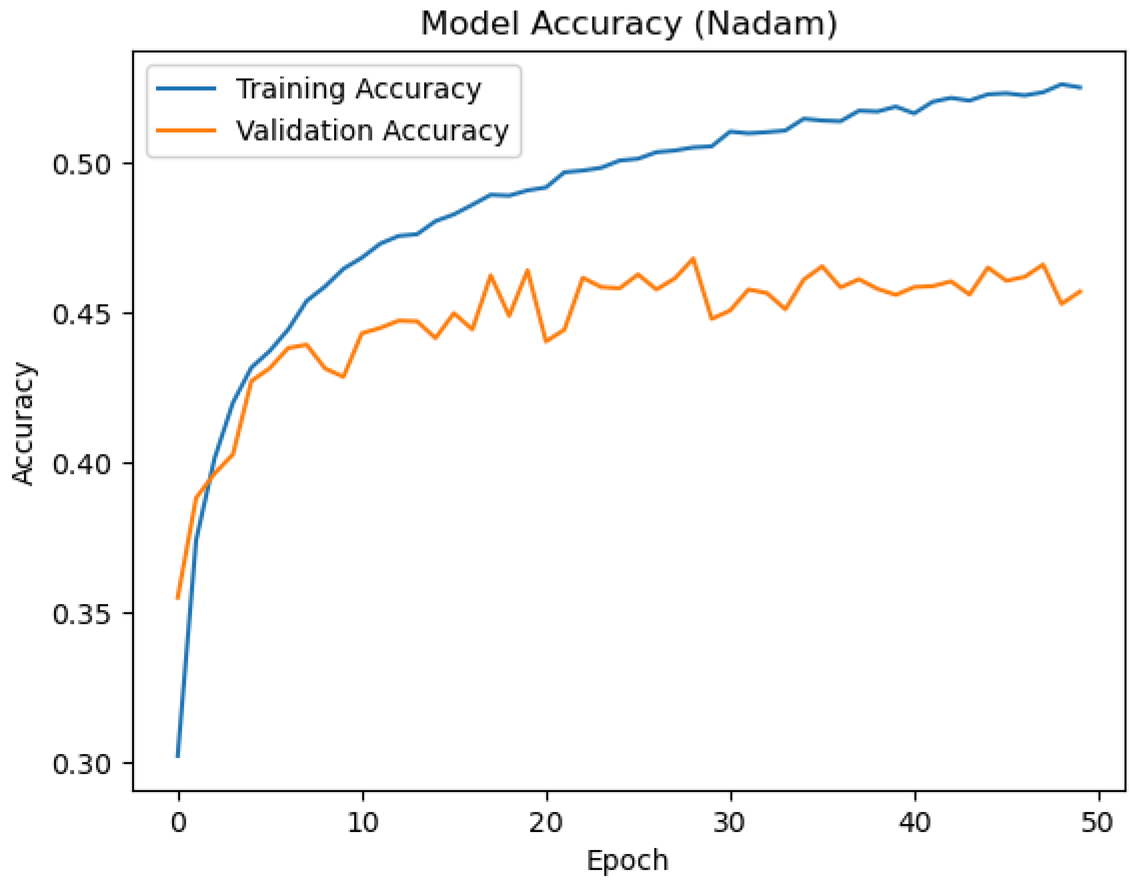
```
625/625 [==============================] - 5s 8ms/step - loss: 1.3588 - accuracy: 0.
5171 - val_loss: 1.5275 - val_accuracy: 0.4610
Epoch 39/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3589 - accuracy: 0.
5168 - val_loss: 1.5529 - val_accuracy: 0.4578
Epoch 40/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3551 - accuracy: 0.
5185 - val_loss: 1.5496 - val_accuracy: 0.4558
Epoch 41/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3549 - accuracy: 0.
5162 - val_loss: 1.5583 - val_accuracy: 0.4584
Epoch 42/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3509 - accuracy: 0.
5201 - val_loss: 1.5414 - val_accuracy: 0.4587
Epoch 43/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3495 - accuracy: 0.
5214 - val_loss: 1.5358 - val_accuracy: 0.4603
Epoch 44/50
625/625 [==============================] - 5s 9ms/step - loss: 1.3474 - accuracy: 0.
5205 - val_loss: 1.5427 - val_accuracy: 0.4559
Epoch 45/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3439 - accuracy: 0.
5226 - val_loss: 1.5326 - val_accuracy: 0.4649
Epoch 46/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3405 - accuracy: 0.
5229 - val_loss: 1.5546 - val_accuracy: 0.4605
Epoch 47/50
625/625 [==============================] - 5s 9ms/step - loss: 1.3402 - accuracy: 0.
5222 - val_loss: 1.5510 - val_accuracy: 0.4618
Epoch 48/50
625/625 [==============================] - 5s 9ms/step - loss: 1.3364 - accuracy: 0.
5233 - val_loss: 1.5320 - val_accuracy: 0.4659
Epoch 49/50
625/625 [==============================] - 5s 8ms/step - loss: 1.3335 - accuracy: 0.
5259 - val_loss: 1.5735 - val_accuracy: 0.4528
Epoch 50/50
625/625 [==============================] - 4s 7ms/step - loss: 1.3329 - accuracy: 0.
5249 - val_loss: 1.5504 - val_accuracy: 0.4569
Training time: 263.21 seconds
```

In [7]:

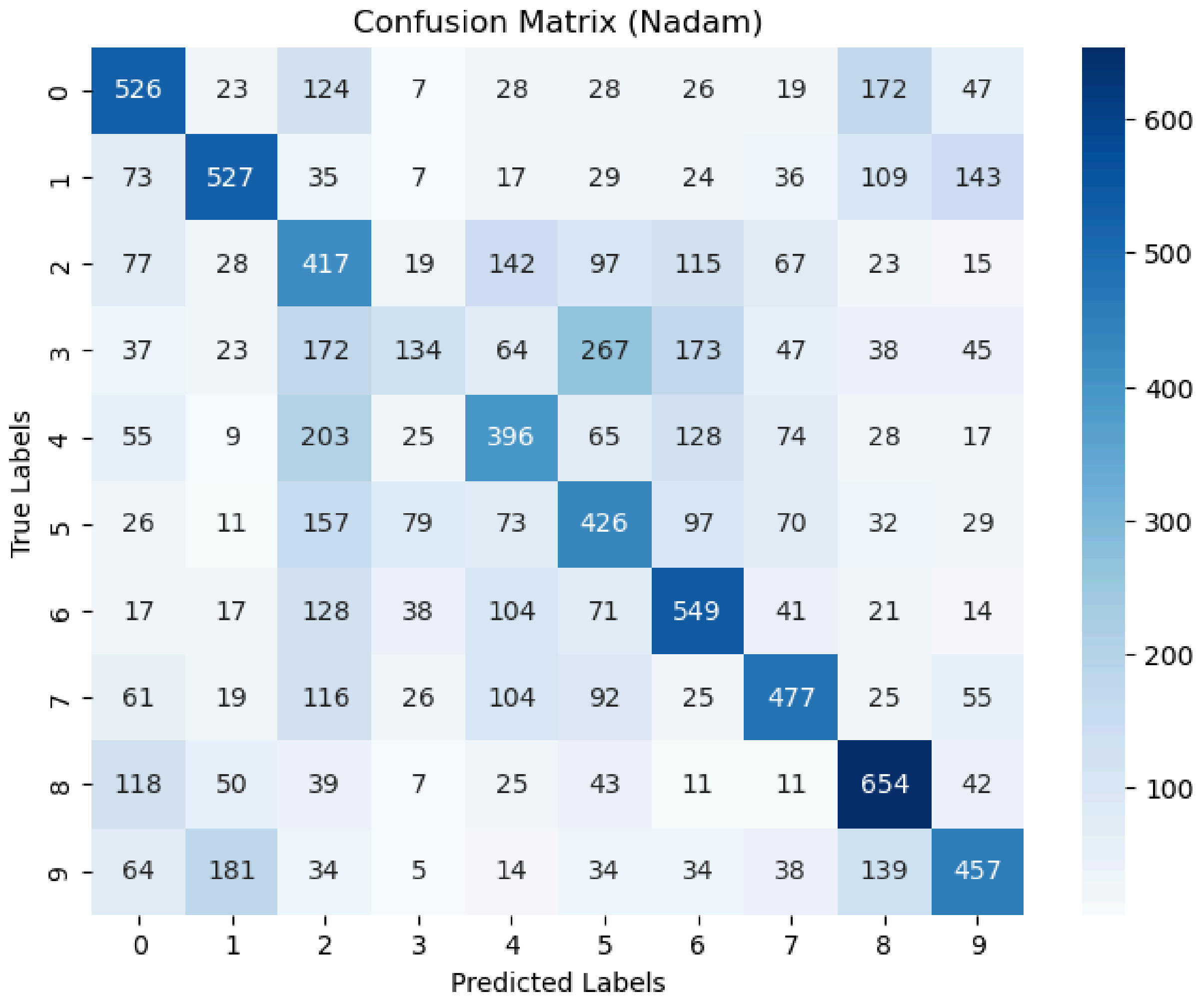
```python
# Plot training and validation accuracy
accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
plt.plot(accuracy, label='Training Accuracy')
plt.plot(val_accuracy, label='Validation Accuracy')
plt.title('Model Accuracy (Nadam)')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

# Confusion Matrix
y_pred = np.argmax(model.predict(X_test), axis=1)
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=np.arange(10), ytick
```

```
plt.title('Confusion Matrix (Nadam)')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

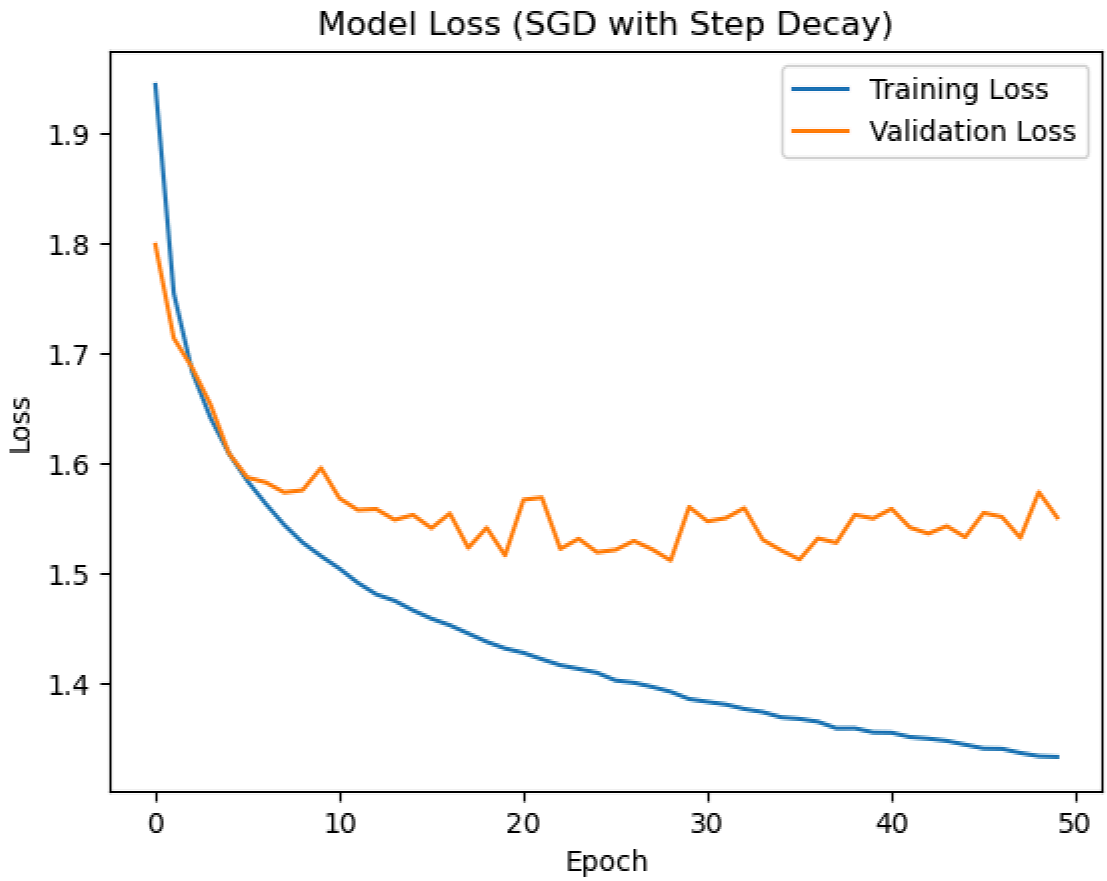



```
313/313 [==============================] - 2s 6ms/step
```



In [6]:

```
# Plot training and validation accuracy
accuracy = history.history['loss']
val_accuracy = history.history['val_loss']
plt.plot(accuracy, label='Training Loss')
plt.plot(val_accuracy, label='Validation Loss')
plt.title('Model Loss (SGD with Step Decay)')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```


Model Loss (SGD with Step Decay)
Training Loss
Validation Loss
Loss
1.9
1.8
1.7
1.6
1.5
1.4
0
10
20
30
40
50
Epoch

```
In [ ]:
```