1. Apple Vision Pro is a mixed-reality headset combining augmented reality (AR) and virtual reality (VR). It provides a high-resolution display, spatial audio, and immersive user experiences, targeting various productivity, entertainment, and communication applications.

https://time.com/collection/best-inventions-2023/6323191/apple-vision-pro/

2. PlayStation VR (PSVR), released in 2016, is a virtual reality headset made for Sony's PlayStation consoles. It gave gamers an affordable way to experience VR, bringing immersive, 360-degree gaming to a wider audience. By working with an already popular gaming system, PSVR became a hit among gamers and helped make VR more well-known, though its use stayed mainly in the gaming world.

On the other hand, Apple Vision Pro takes things further by combining virtual reality (VR) and augmented reality (AR). While PSVR is mainly for gaming, Vision Pro is built for much more, including entertainment, work, and communication. Apple Vision Pro is likely to reach a larger group of users, not just gamers but also professionals and everyday people who want new, immersive ways to interact with the digital world. Unlike PSVR's focus on gaming, Vision Pro aims to be useful in many areas of life, making it appealing to a broader audience. https://www.playstation.com/en-us/ps-vr/

3. PSVR - https://www.statista.com/statistics/987693/psvr-unit-sales/

Vision pro - https://www.statista.com/statistics/1398458/apple-vision-pro-shipments/

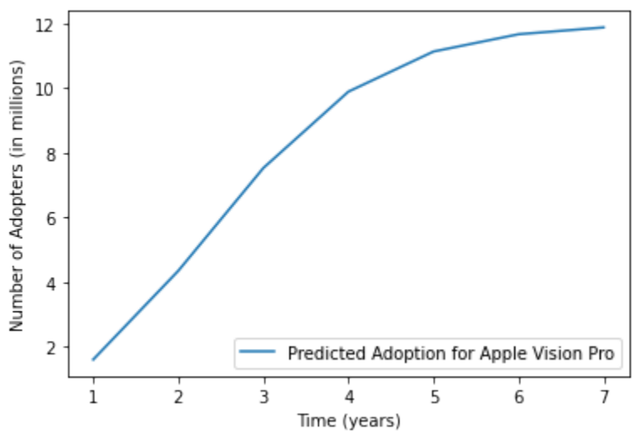
In [22]:

```
#4
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt # type: ignore

def bass_model(t, p, q, M):
    exp_term = np.exp(-(p + q) * t)
    F_t = (1 - exp_term)/(1 + (q / p) * exp_term)
    return M * F_t

time = np.array([4, 14, 22, 29, 39])  #months since launch
adoption_data = np.array([0.92, 2.00, 3.00, 4.20, 5.00])

best_params = (0, 0, 0)
min_error = float('inf')

#going through possible values for p, q, and M
for p in np.arange(0.01, 0.1, 0.01):
    for q in np.arange(0.1, 1.0, 0.1):
        for M in np.arange(5, 10, 1):
            predictions = bass_model(time, p, q, M)
            error = np.sum((adoption_data - predictions) ** 2)
            if error <= min_error:
                best_params = (p, q, M)

p, q, M = best_params
print(f"Estimated Bass Model Parameters: p = {p}, q = {q}, M = {M}")
```

```
Estimated Bass Model Parameters: p = 0.09, q = 0.9, M = 9
```

In [19]:

```
#5
time_future = np.arange(1, 8, 1)  #8 years
#for less than 8 years the total result was significantly lower than 12 million
M_vision_pro = 12  #market potential 12 million (source in part 3)
vision_pro_prediction = bass_model(time_future, p, q, M_vision_pro)

plt.plot(time_future, vision_pro_prediction, label="Predicted Adoption for Apple Vision Pro")
plt.xlabel('Time (years)')
plt.ylabel('Number of Adopters (in millions)')
plt.legend()
plt.show()
```



1. Both are internationally recognized brands. Their strong reputations mean they can attract users across various markets. They won't have competition because there are no competing products in other countries that offer the same level of compatibility and quality.

In [20]:

```
#7

for year in range(1, len(vision_pro_prediction)):
    new_adopters = round(vision_pro_prediction[year] - vision_pro_prediction[year - 1], 2)
    print(f"Year {year}: {new_adopters} million new adopters")
```

```
Year 1: 2.75 million new adopters
Year 2: 3.18 million new adopters
Year 3: 2.36 million new adopters
Year 4: 1.24 million new adopters
Year 5: 0.53 million new adopters
Year 6: 0.21 million new adopters
```