Sources:

Key References and Inspirations:

1. **Research Paper: "Lagrangian Neural Networks" by Cranmer et al., 2020**
   - **DOI:** 10.48550/arXiv.2003.04630
   - This foundational paper introduces the concept of LNNs, explaining how to structure a neural network to learn dynamics by leveraging the Lagrangian formulation.
   - Key ideas include using the Euler-Lagrange equations to compute dynamics and enforcing physical consistency during training.
2. **Lagrangian Mechanics Textbooks**
   - Standard physics resources such as:
     - *"Classical Mechanics"* by Herbert Goldstein
     - *"Classical Dynamics of Particles and Systems"* by Thornton and Marion
   - These provide a detailed understanding of Lagrangian mechanics, including deriving equations of motion for simple systems like pendulums and sliding blocks.
3. **PyTorch Documentation**
   - Official PyTorch documentation on:
     - Automatic Differentiation (autograd)
     - Neural Networks API
   - PyTorch's autograd is central to computing derivatives needed for Lagrangian dynamics.
4. **Machine Learning for Physics**
   - Tutorials and blog posts discussing physics-informed neural networks (PINNs) and their application to classical mechanics problems.
   - Example: Blogs from DeepAI, Medium, or GitHub repositories focused on physics-informed ML.
5. **GitHub Projects**

- Open-source implementations of LNNs and Hamiltonian Neural Networks (HNNs), such as those found in repositories like:
  - Cranmer's GitHub Repository for LNNs