

# Fatima Bazzoun

📍 Beirut, Lebanon

✉️ fmb26@mail.aub.edu

LinkedIn

GitHub

## Education

- **American University of Beirut (AUB)** Sep. 2023 – Jun. 2027 (Expected)  
*Bachelor's in Computer and Communications Engineering*  
Full **CNRS** Scholarship (ranked 3rd nationwide in the Lebanese official exams).  
Dean's Honor List every semester. GPA: **4.0/4.0**.  
Member, **AUB Quantum Club** — Social Media Team.

## Experience

- **Teaching Assistant, American University of Beirut** Jan. 2025 – Present
  - Led labs and review sessions for **Data Structures & Algorithms** for 100+ students across 3 semesters, coordinating with 3 professors; taught **C++ implementations** of core data structures (arrays, linked lists, trees, heaps, graphs), recursion, and asymptotic analysis.
  - Designed competitive-programming style exercises, delivered live coding demos, and held structured weekly office hours; supported student Q&A via course forum and email.
  - Developed grading rubrics; graded programming assignments and homework with written feedback; managed regrade requests fairly and promptly; proctored midterms/finals while following academic integrity policies and procedures.
- **VIP Program, American University of Beirut** Jan. 2025 – Present
  - Designing quantizers for  $\alpha$ -stable sources by exploring alternative distortion measures, deriving Lloyd–Max-style optimality conditions, and evaluating rate–distortion trade-offs.
  - Built an interactive geospatial map (**Django + Leaflet**) to visualize model outputs: upload → inference → SQLite persistence → map display with dynamic pins.
  - Implemented user authentication (login/signup) and personalized views; surveyed user reviews and stack options (frameworks, data models, DB choices) for a plant-app concept to guide design decisions.
- **Summer Intern, GeoAI Research Group** May 2025 – Aug. 2025
  - Prototyped vehicle motion detection for SAR/ViSAR stacks using symmetric frame differences, median background modeling, and size/percentile filters; evaluated optional BM3D denoising.
  - Researched and applied **PSInSAR/QPSInSAR** (denser points across event timelines) and **PolSAR/PolInSAR** (scattering semantics) for disaster-response and time-sensitive event analysis.
  - Ran an end-to-end InSAR analysis on a real AOI.
  - Conducted literature reviews (dockless bike demand from satellite data; ViSAR + shadow vehicle-velocity detection) and completed NASA introductory remote-sensing onboarding material.

## Projects

- **Machine Learning for NP Problems** Sept. 2025 – Present
  - Investigated ML-based approaches for NP-hard graph problems (*Longest Path, Hamiltonian Cycle*) using supervised learning on synthetic directed graphs.
  - Designed a per-node feature extraction pipeline (SCC statistics, in/out-degree, local path-length estimates, etc.); fed node features into a PyTorch DNN to predict the longest path starting from each node.
  - Generated a large synthetic dataset of graphs with node-level feature vectors and ground-truth labels obtained from exact solvers for small and moderate graph sizes; exported to CSV and subsampled to **100k** examples for training.
  - Trained and tuned a deep neural network on node-level features and benchmarked performance against classical graph-theoretic heuristics.
  - Achieved an average performance ratio of **0.877** compared to a baseline greedy heuristic (selecting the vertex with the most unvisited neighbors); on the same 100 test graphs, obtained an average path length of **18.87** versus **9.69** for the baseline, outperforming it in **100/100** cases.
- **Multiplayer Car Racing Game** Mar. 2025 – Apr. 2025
  - Engineered a Python real-time multiplayer game using **TCP/UDP sockets** (hybrid client–server + P2P), multithreaded state replication, **SQLite** persistence, and synchronized multimedia sharing.
- **Chess Rules Engine** Jan. 2024
  - Built a C++ object-oriented chess rules core with a polymorphic piece hierarchy and board-state model.
  - Implemented legal-move validation using state simulation/reversion to prevent self-check, plus attack-square-based check detection.

## Skills

- **Programming:** Python, C++, Verilog, Assembly, MATLAB
- **Frameworks & Libraries:** Django, NumPy, Pandas, SciPy, Matplotlib, PyQt, Pygame, Leaflet, Pytorch
- **Tools & Platforms:** Git, GitHub, SQLite, Jupyter Notebook, SNAP (ESA), QGIS, Copernicus