
JOIN OUR TEAM

R.02: Intern - ML Engineering & Prototype Integration

Research · Guwahati · Hybrid · Full-time

Function	Location	Commitment	Compensation
ML Engineering & Prototype Integration	Guwahati (Hybrid)	Full-time	Stipend + up to 30% incentive

ABOUT THE ROLE

QELP AI is developing a multimodal neurocognitive AI platform for understanding human cognitive and performance states from heterogeneous data.

We are looking for an ML Engineering & Prototype Integration Intern to work at the interface between machine-learning research and working software systems.

You will implement and evaluate models under the guidance of the technical lead, translate research outputs into reproducible inference pipelines, and integrate selected models into usable software prototypes.

This is a hands-on engineering role. You will be expected to work with real datasets, train and evaluate models, investigate failures, build inference workflows and turn validated components into runnable prototypes.

WHAT SUCCESS LOOKS LIKE

By the end of the initial engagement, another researcher or engineer should be able to reproduce the model training and evaluation workflow, load the selected model, run inference on the defined input data, and use the resulting prototype without relying on undocumented steps or the original developer.

WHAT WE OFFER

- Compensation: Competitive stipend for the engagement.
- Performance incentive: Up to 30% of the stipend, based on performance and achievement of defined deliverables.

- Employment opportunity: High-performing candidates may be considered for formal employment with QELP AI, with compensation aligned to the role, relevant experience and grade.
-

— **ROLE OBJECTIVE**

Implement and evaluate machine-learning models under the technical lead's guidance and integrate the selected model into a usable, reproducible and documented software prototype.

— **WHAT YOU WILL WORK ON**

- Implement and adapt machine-learning models for defined research problems.
 - Prepare training and evaluation workflows from structured and time-series datasets.
 - Train, validate and compare candidate models using appropriate evaluation methodologies.
 - Save, version and load trained model checkpoints reliably.
 - Develop reproducible inference pipelines from data input to model prediction.
 - Investigate model errors, unexpected behaviour and performance degradation.
 - Identify and address issues such as overfitting, data leakage and class imbalance.
 - Implement appropriate evaluation metrics and maintain clear experiment records.
 - Build interfaces or APIs for interacting with trained models.
 - Integrate selected models into working software prototypes.
 - Test prototype functionality across defined input conditions and failure cases.
 - Work with the research and technical teams to translate experimental requirements into implementable systems.
 - Maintain clear documentation of code, dependencies, model versions, experiments and deployment procedures.
-

— **ESSENTIAL QUALIFICATIONS**

Academic Background

B.E./B.Tech. with relevant experience, M.E./M.Tech., or M.Sc. in one of the following or a closely related discipline:

- Computer Science
- Artificial Intelligence / Machine Learning
- Data Science
- Electronics
- Computer Engineering

- Information Technology
- Applied Mathematics
- Statistics

Candidates must meet the applicable eligibility requirements of the appointing institution.

Experience

Prior project, research or professional experience involving machine learning, model development or software-based ML applications is preferred.

— ESSENTIAL PRACTICAL SKILLS

- Strong Python programming.
- Practical experience with scikit-learn and PyTorch or TensorFlow.
- Experience training, saving, loading and evaluating machine-learning models.
- Understanding of: overfitting, data leakage, class imbalance, train/validation/test methodology, appropriate evaluation metrics.
- Experience building a working ML interface or API using Streamlit, Gradio, FastAPI or an equivalent framework.
- Familiarity with Git, dependency management and debugging.
- Understanding of basic software testing and reproducible development practices.

Minimum Evidence of Competence

Candidates should be able to demonstrate at least one completed project showing the path from data input, through model processing and prediction, to a runnable application or API.

Candidates should be prepared to walk through their architecture, code, model, evaluation methodology and engineering decisions during the interview.

— DESIRABLE SKILLS

Experience with one or more of:

- Time-series modelling
- Multimodal machine learning or data fusion
- Sensor or physiological data
- Linux
- Docker
- Experiment tracking
- Model/version management

- Inference optimization
- Model deployment
- API development
- Basic CI/CD or automated testing

Candidates do not need experience across all areas. Strong ML fundamentals combined with evidence of building and integrating a working system are valuable.

— THREE-MONTH RESEARCH & ENGINEERING DELIVERABLES

01 Reproducible Training Pipeline

Clean, modular and reproducible training code covering data preparation, model training, validation and evaluation.

02 Evaluated Model Checkpoints

Versioned model checkpoints accompanied by documented training configuration, evaluation results and relevant experiment records.

03 Inference Pipeline

A reproducible inference workflow that accepts defined inputs, loads the appropriate model and produces the required predictions or outputs.

04 Model Evaluation Record

A documented record of model comparisons, evaluation metrics, experimental conditions, limitations and key findings.

05 Working Demonstration Application

A functional prototype that allows a user to provide defined inputs and interact with the selected model through a usable interface or API.

06 Integration & Testing

Documented testing of the integrated prototype across representative inputs, expected operating conditions and identified failure cases.

07 Engineering Documentation & Audit Trail

A complete, version-controlled record of code, dependencies, data and model versions, configuration, experiments, evaluation results, implementation decisions, deviations and known limitations, enabling another researcher or engineer to reproduce the workflow and independently assess the integrity of the implementation and results.

08 Setup & Handover

Clear setup, execution and handover documentation enabling the technical team to install the required environment, reproduce the prototype, run inference and continue development without dependence on the original developer.

VALIDATED BY

Meghali Deka

APPROVED BY

Dr. Doli Hazarika