SmartWatt

D5: Anya Bindra, Erika Ramirez, Maya Doshi
18-500 Capstone Design, Spring 2025
Electrical and Computer Engineering Department
Carnegie Mellon University

https://course.ece.cmu.edu/~ece500/projects/s25-teamd5/

Product Pitch

For homeowners with solar panels, **SmartWatt** cuts peak-time electricity costs by up to **15%** in high-load scenarios and improves solar self-consumption by **~30%** compared to manual scheduling. The system uses **linear programming** to generate daily appliance usage schedules that minimize power expenses while meeting user-defined constraints. A web-based low latency **UI dashboard** provides real-time energy data and user controls. The frontend displays appliance-level insights, including usage trends, solar power contribution, and personalized recommendations for optimal on/off times.

System Description

- **Electrical Components**: Appliances connected via ESP32 for remote control from the dashboard.
- **Sensors**: Power, voltage and current sensors collect real-time consumption data for each device.
- **Home Assistant**: Local controller and data aggregator, interfacing with both the sensors and devices via MQTT

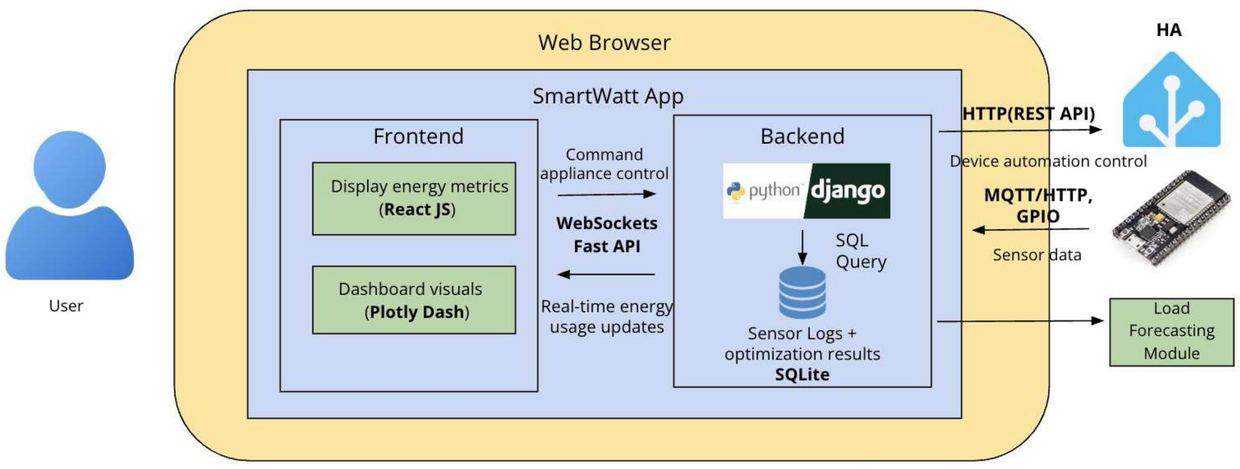
Backend (FastAPI + Python + OpenAI):

- Host linear programming optimizer that schedules appliance operations to minimize cost based on solar and grid prices.
- Runs a regression ML model for forecasting solar generation and dynamic electricity pricing from the grid.

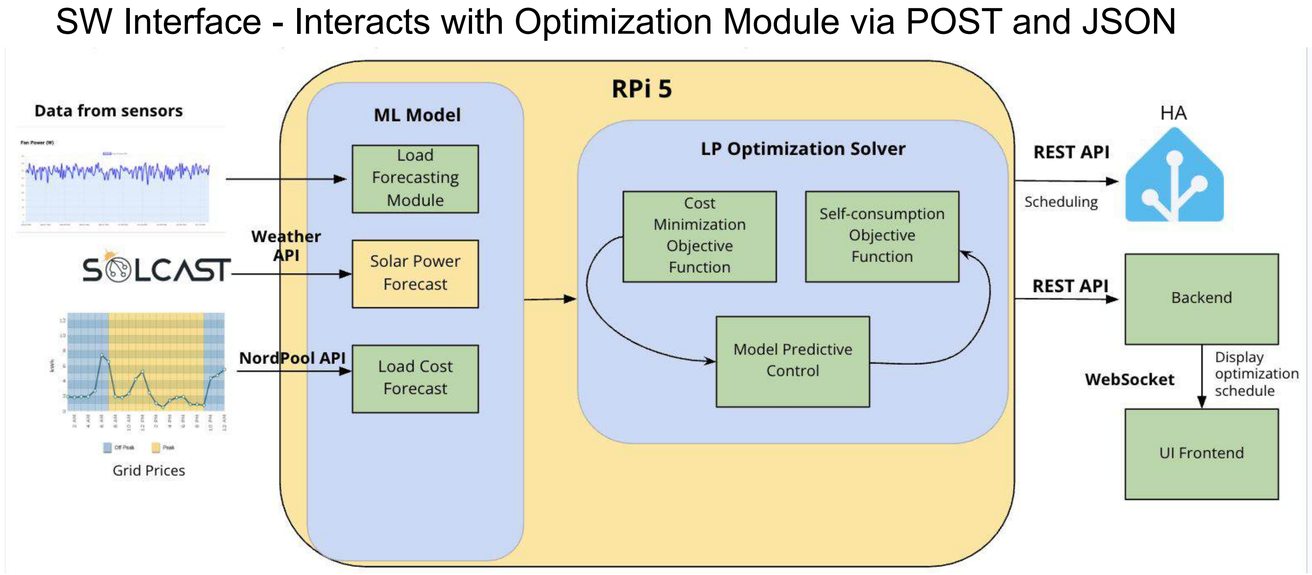
Frontend Dashboard:

- Built with Jinja2 and React providing users with live consumption insights, schedule visualizations, and manual override controls.

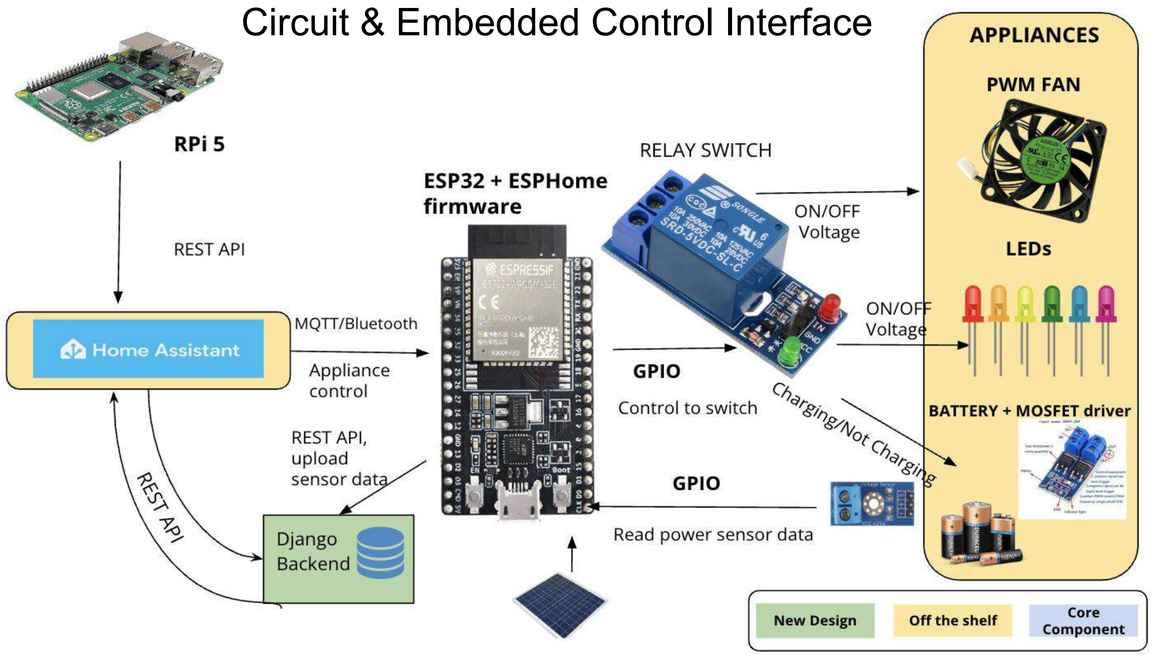
System Architecture

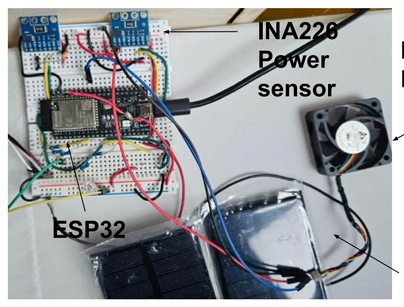


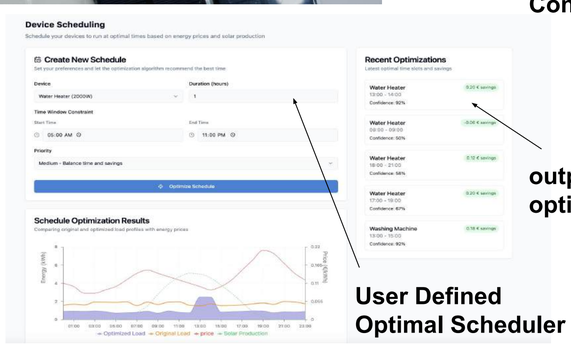
SW Interface - Interacts with Optimization Module via POST and JSON

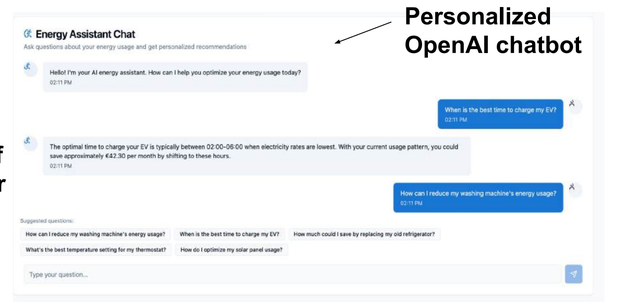


Circuit & Embedded Control Interface



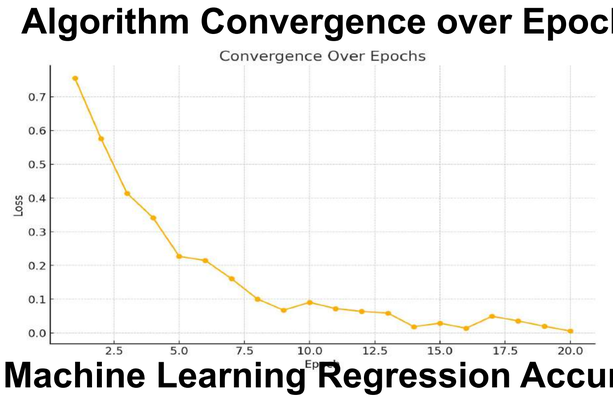


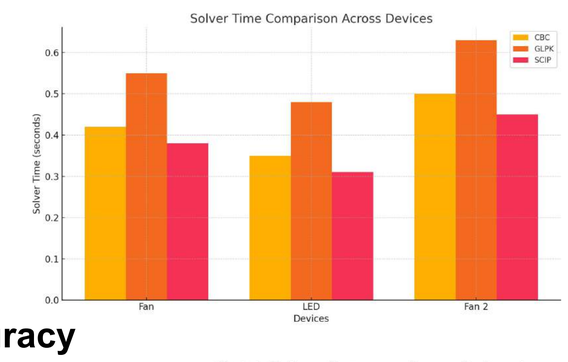




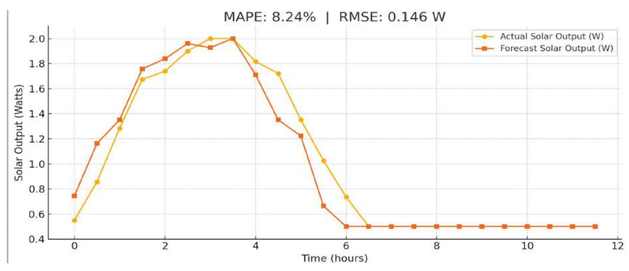
System Evaluation

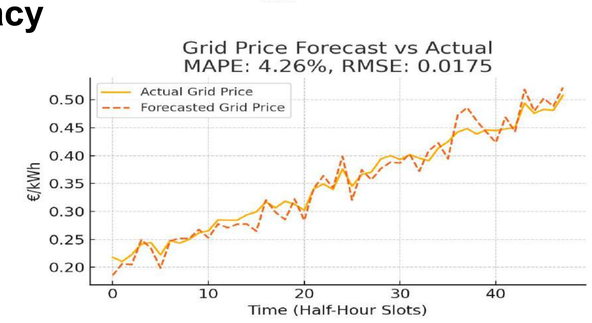
Algorithm Convergence over Epochs and Wall Clock Time





Machine Learning Regression Accuracy





Use-Case Requirements:

Functionality	Measurement	Test Input	Test Output
Cost Reduction via Optimization (15% target)	Monthly Energy Cost change (%)	Run devices with and without scheduling (assign random times) over 7 days	**17%** lower cost using SmartWatt optimizer
Load Shifting Efficiency (20% target)	% of load shifted to low cost hours, high solar output hours	Power Profile under default vd optimized schedule	**24%** of consumption moved to hours with price < daily median
Device Control + Actuation Latency (3s target)	**2s** delay	POST request to endpoint with device action and scheduling	All control signals result in correct ON/OFF state. The latency is **1.4s**
Backend Data Fetching Accuracy - Power readings from INA226 sensor	Measure current and power within **98%** of actual reading.	Fetch voltage, current, and power from INA226 sensor	**0.5%** power calibration error (from measured V and I values)

Conclusions & Additional Information

We developed a smart home system that automates energy management. Integrating device control with backend scheduling and training scalable ML models presented challenges, but taught us how to effectively synchronize hardware with software. Stress testing with flat and spiky data exposed edge cases and API logging was critical for debugging. Asynchronous task queuing improved UI responsiveness by preventing blocking during optimization.We also gained experience in user-centered design. SmartWatt can be extended with battery, wind and geothermal power integration.

Electrical & Computer ENGINEERING
Carnegie Mellon