

Project 1 Spec

Title: Productivity Proxy Prediction API for Students

1. User & Decision

This API is designed for students who want to manage their study schedules more effectively. As a student, predicting *"How productive will I be today?"* can help decide whether to push through assignments, take breaks, or reduce distractions. The goal is not to perfectly measure true productivity, but to provide a **proxy productivity score** based on lifestyle inputs. Even with limited resources (free-tier cloud services), this API should deliver useful insights without excessive cost.

2. Target & Horizon

- **Target:** A **daily productivity proxy score** (0-10) based on self-reported stress, attention span, or productivity scores from open datasets.
 - **Horizon:** 12-24 hours (today or the next day).
 - **Justification:** Productivity cannot be objectively measured at scale, but proxy labels offer a practical and ethical way to approximate it.
-

3. Features (No Leakage)

- **Core inputs:**
 - **sleep_hours** (previous night)
 - **caffeine_mg** (consumed today)
 - **screen_time_hours** (today or previous day)
 - **Optional:** bedtime, notification counts.
 - **No leakage:** All features are past or current values; no future data is used.
-

4. Baseline → Model Plan

- **Baseline:** Rule-based model.
 - sleep < 6h → lower score
 - caffeine 0-200mg → slightly higher score; > 400mg → lower score
 - screen_time > 6h → lower score
 - **Model:** Lightweight regression (Linear Regression).
 - **Deployment:** Model exported to ONNX for small size and fast inference in serverless environments.
-

5. Metrics, SLA, and Cost

- **Metrics:** RMSE / MAE (continuous), Brier score (probabilities).
- **SLA:**
 - p95 latency ≤ 300ms

- cold start < 1s
 - availability ≥ 99%
 - **Cost envelope:** service remains within serverless **free-tier**; target ≈ **\$0/month** under normal student load; **monitored with synthetic probes every 1 minute @ ~1k req/min**; on viral spikes (**e.g., 50k req/h**) automatically **degrade to baseline (rules-only)** to cap compute and avoid overage; **stateless** by design (raw=0d; aggregates ≤30d).
 - **Cost constraints (student reality):**
 - Free-tier limits on AWS Lambda or Vercel (millions of free invocations/month).
 - No GPU usage; CPU-only inference.
 - No paid monitoring; instead, lightweight logging.
 - **Viral spike survival:** In case of 50k req/h, switch to rule-based fallback that runs instantly with no compute cost.
-

6. API Sketch

Endpoint: `POST /v1/predict`

Request:

```
{
  "sleep_hours": 6.2,
  "caffeine_mg": 120,
  "screen_time_hours": 5.5
}
```

Response:

```
{
  "proxy_productivity_score": 6.8,
  "uncertainty": 0.2,
  "risk_factors": ["late_bedtime", "high_screen_time"],
  "suggestion": "Reducing screen time by 1h could increase score by ~0.5"
}
```

7. Privacy, Ethics, Reciprocity (PIA excerpt)

(🔗 See full PIA: <https://github.com/lna-nk-lee/productivity-proxy-prediction-API/blob/main/PIA.pdf>)

- **Data Collected:** sleep_hours, caffeine_mg, screen_time_hours (lifestyle only, no PII)
- **Purpose:** predict proxy productivity score (0–10)
- **Retention:** none for raw inputs (0 days); aggregates kept ≤ 30 days
- **Access:** developer-only; no third-party sharing
- **Guardrails:** input validation (reject outliers), jitter/noise for aggregates, opt-in telemetry only

- **Disclaimer:** This API predicts a *proxy productivity score*, not medical or clinical outcomes.
-

8. Architecture & Feasibility

Architecture:

- Client → API Gateway → Serverless (AWS Lambda or Vercel) → ONNX model

Degrade mode:

- If free-tier compute limits are reached or latency spikes, switch to baseline rule predictions.

Trade-offs:

- Serverless (cheap, scales automatically) vs container (expensive, not feasible for free-tier students).
 - For students, serverless is the only realistic option; cold starts are acceptable if rare.
-

9. Risks & Mitigations

- **Risk:** Dataset labels (stress, attention) are proxies, not true productivity.
 - *Mitigation:* Acknowledge in documentation; use assumption audit.
 - **Risk:** Free-tier limits exceeded.
 - *Mitigation:* Automatic degrade mode to baseline rules.
 - **Risk:** Noisy self-report features.
 - *Mitigation:* Provide uncertainty values; clip extreme inputs.
-

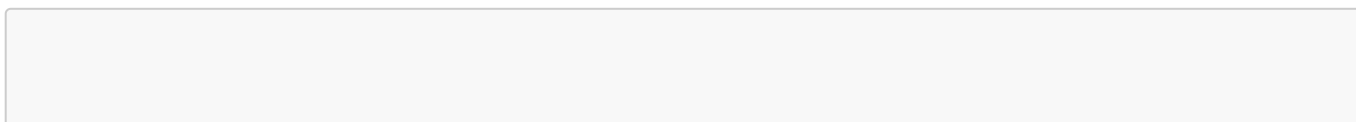
10. Measurement Plan

- **Minimal experiment:**
 - Train/test split on Kaggle dataset with productivity-related proxies.
 - Compare baseline RMSE vs model RMSE.
 - **SLA test:**
 - **Synthetic probes every 1 minute @ ~1k req/min** to measure p95 latency & cold start.
 - Simulate free-tier limits; measure latency under ~1k req/min.
 - Test fallback baseline mode for stability.
-

11. Evolution & Evidence

- Reframed anxiety → productivity proxy.
- Added non-linear caffeine effect.
- Chose serverless + fallback over containers.
- Differentiated from CalmCast (inputs & horizon).

Git Evidence:



```
commit eb070297f21d44ac3d88a8c445f18b84486235ae
Date:   Sun Sep 28 11:12:01 2025 -0700

    add diagram and evaluation plan

commit 3238461d994ec2d35a3fea427202d484d9b55f57
Date:   Sun Sep 28 12:06:59 2025 -0600

    add API sketch to README

commit 1e0df43f81a119d47d663d983fe5788e922c79b4
Date:   Sat Sep 27 16:09:08 2025 -0600

    Initial commit
```

◆ Conclusion

This API provides a realistic, low-cost solution for predicting daily productivity proxies based on lifestyle inputs. It acknowledges the limits of using stress and attention span as proxies, while still offering practical value to students. The architecture is feasible under free-tier constraints, with fallback strategies to handle viral load. The design covers rubric requirements in problem framing, specification clarity, privacy/ethics, baseline evaluation, architecture, risks, and evolution.

Architecture Diagram & Minimal Evaluation Plan

Architecture Diagram (Mermaid)

```
flowchart LR
    A[Client] -->|/predict| B[API Gateway]
    B --> C[Compute: Serverless Function]
    C --> D[(Model: ONNX)]
    C --> E[(Fallback: Rule-based Baseline)]
    C --> F[Observability]

    subgraph Guardrails
        G[Retention: raw=0d, aggregates ≤30d]
        H[Minimal telemetry + jitter if shared]
        I[SLA target: p95 latency ≤300ms; cold start <1s]
    end

    F -.-> Guardrails
```

Minimal Evaluation Plan

Baseline

- Rule-based: sleep < 6h → score↓, caffeine > 400mg → score↓, screen_time > 6h → score↓

Model

- Lightweight regression (Linear/XGBoost), exported to ONNX

Metrics

- RMSE / MAE (continuous proxy score)
- SLA metric: p95 latency ≤ 300ms, cold start < 1s

Evaluation

- Train/test split on open dataset (proxy labels: stress, attention span, productivity)
- Compare RMSE of baseline vs model
- **SLA measurement: synthetic probes every 1 minute @ ~1k req/min**
- Load test: simulate 1k req/min (fits free-tier), test viral spike (50k req/h) with fallback

Cost envelope

- **Within free-tier (~\$0/month** under normal student load); measured alongside SLA probes; if projected overage, automatically **degrade to baseline** and reduce telemetry frequency; stateless inputs (raw=0d; aggregates ≤30d).

Insight Memo, Assumption Audit, Socratic Log, Git Evidence

Insight Memo (3 Key Insights)

1. **Proxy ≠ Productivity:** Productivity cannot be measured directly. Using stress, attention span, and self-reported scores as proxies makes the target tractable but requires honesty about limitations.
2. **Free-tier Constraints Shape Architecture:** Designing for AWS Lambda/Vercel with fallback rules ensures the system remains viable for students without paid infrastructure. Cost limitations fundamentally influenced design choices.
3. **Baseline vs Model Trade-off:** Rule-based baselines are explainable and resilient at scale, while lightweight ML models add nuance. Maintaining both supports reliability under viral load.

Assumption Audit

- **Sleep hours ↑ → Productivity proxy score ↑** (assumed positive linear relation).
- **Caffeine has non-linear effect:** moderate intake improves score; excessive intake lowers score.
- **Screen time hours ↑ → Productivity proxy score ↓** (assumed negative correlation).
- **Inputs are self-reported and noisy:** must include uncertainty estimates and rejection of outliers.

Socratic Log References

Q1: Could I predict anxiety disorder from sleep, caffeine, and screen time?

A1: That would raise serious ethical issues (medical diagnosis, misuse risk).

➔ **Insight:** Avoid framing as clinical/medical prediction.

Q2: What if I reframe it as productivity prediction?

A2: Productivity is vague, but you can use *stress/attention span/productivity scores* as proxies.

➡ **Insight:** Define the target explicitly as a **proxy productivity score**.

Q3: Isn't caffeine always linked to higher productivity?

A3: Not directly. Moderate caffeine can improve focus, but excessive caffeine worsens sleep and productivity.

➡ **Insight:** Capture caffeine's **non-linear effect** in baseline rules.

Q4: Should I deploy the model in containers instead of serverless?

A4: Containers are stable but exceed free-tier constraints. For students, serverless + fallback is more practical.

➡ **Insight:** Architecture shaped by **free-tier cost guardrails**.

Q5: Is this project too similar to CalmCast from the idea list?

A5: CalmCast uses smartphone activity logs, while yours uses lifestyle factors (sleep, caffeine, screen time). Different framing, but important to clarify.

➡ **Insight:** Explicitly highlight difference from prior idea to avoid overlap.

Git Evidence

```
commit 5b796fedd57a3dfd929c791aa9a2a2cd6aaa11ac (HEAD -> main,
origin/main, origin/HEAD)
```

```
Author: Ina Lee <inalee1208@gmail.com>
```

```
Date:    Sun Sep 28 16:27:07 2025 -0700
```

```
    add PIA as a separate file
```

```
commit 506fb7d67c515ff03bd1841800b0bd626a3231f3
```

```
Author: Ina Lee <inalee1208@gmail.com>
```

```
Date:    Sun Sep 28 17:02:06 2025 -0600
```

```
    update baseline model information
```

```
commit 89f80641f1e5a2867beef1b586d06adc2cfb0fbb
```

```
Author: Ina Lee <inalee1208@gmail.com>
```

```
Date:    Sun Sep 28 16:36:06 2025 -0600
```

```
    shorten spec
```

```
commit f75644382458e803683f189554821ba2ff0c63bc
```

```
Author: Ina Lee <inalee1208@gmail.com>
```

```
Date:    Sun Sep 28 12:53:47 2025 -0600
```

```
    format README
```

```
commit 121bfd6180a0f34f7bc796ed5d6db17d75e104ef
```

```
Author: Ina Lee <inalee1208@gmail.com>
```

```
Date:    Sun Sep 28 11:13:32 2025 -0700
```

```
    add PIA excerpt and Telemetry Decision matrix
```

```
commit eb070297f21d44ac3d88a8c445f18b84486235ae
```

Author: Ina Lee <inalee1208@gmail.com>

Date: Sun Sep 28 11:12:01 2025 -0700

add diagram and evaluation plan

commit 3238461d994ec2d35a3fea427202d484d9b55f57

Author: Ina Lee <inalee1208@gmail.com>

Date: Sun Sep 28 12:06:59 2025 -0600

add API sketch to README

commit 1951413c5e81148907d31d613b8579fc6275aa2d

Author: Ina Lee <inalee1208@gmail.com>

Date: Sat Sep 27 16:27:11 2025 -0600

update README.md

commit 1e0df43f81a119d47d663d983fe5788e922c79b4

Author: Ina <55104701+Ina-nk-lee@users.noreply.github.com>

Date: Sat Sep 27 16:09:08 2025 -0600

Initial commit