

Final Problem Statement

T1_G20_4InLine

Group_ID	T1_G20
Group_Name	4InLine
Group Leader	Aakash Raju Mohole
Faculty Mentor	Mrs. Tejashri V Deokar
Department	Data Science
Final Problem Assigned	Problem 14. Building a Mental State Monitoring Dashboard

Detailed Final Problem Statement :

Problem ID - 14. Building a Mental State Monitoring Dashboard

Goal: Visualize real-time predictions of mental workload, accuracy probability, and emotion transitions.

Goal: Create a **real-time interactive dashboard** to visualize:

- **Mental workload predictions** (e.g., low/medium/high)
- **Accuracy probability** (likelihood of correct response)
- **Emotion transitions** (e.g., engaged → confused → neutral)

Tools: Streamlit, Dash, Plotly; use pretrained models.

- **Frontend/UI:** Streamlit / Dash (Python-based), Plotly for visualization.
- **Backend/Models:** Pretrained models for workload, accuracy, and emotion (loaded as `.pkl` or `.pt`).
- **Data stream:** Simulated live data (EEG, GSR, Eye, Facial)

Sample Steps for Problem ID - 14.

STEP 1: Define Inputs & Outputs

1.1 Inputs

- **Signals:** EEG, GSR, Eye-tracking, Facial expression features.
- **Model Outputs:**
 - `workload_model` → workload level
 - `accuracy_model` → probability of correct answer
 - `emotion_model` → current emotion state

1.2 Outputs (dashboard visualizations)

- Gauge/indicator for **mental workload**
- Line plot (time-series) of **accuracy probability**
- Sankey/sequence chart for **emotion transitions over time**

- (Optional) Participant summary panel: latest predictions, confidence scores
-

STEP 2: Backend Setup

2.1 Load Pretrained Models

- Load `.pkl` (sklearn/XGBoost) or `.pt` (PyTorch/TensorFlow).
Wrap each model in a predictor function:

```
def predict_workload(eeg_features):  
    return workload_model.predict([eeg_features])[0]  
  
def predict_accuracy(multimodal_features):  
    return accuracy_model.predict_proba([multimodal_features])[0,1]  
  
def predict_emotion(facial_features, eeg_features):  
    return emotion_model.predict([facial_features + eeg_features])[0]
```

2.2 Simulated

- **Simulated:** replay past session CSVs in real-time (e.g., 1 sample/sec).
-

STEP 3: Dashboard Design (UI Components)

3.1 Mental Workload Panel

- **Gauge chart** (Plotly) with levels (Low/Med/High).
Update every second with the latest prediction.

3.2 Accuracy Probability Panel

- **Line chart** showing predicted probability across time.
Add threshold line (e.g., 0.5) to visualize confidence.

3.3 Emotion Transitions Panel

- **Sankey diagram:** flow of emotion states across time blocks.
Alternative: timeline bar chart where color = emotion.

3.4 Participant Summary Panel

- Last N seconds snapshot:
Current workload, accuracy, probability, and emotion.
 - Confidence values displayed as percentages.
-

STEP 4: Implementation with Streamlit/Dash

15. **Set up environment and dependencies** → Install Streamlit/Dash, Plotly, and ML libraries.
16. **Load pretrained models** → Workload, accuracy, and emotion predictors.
17. **Prepare data stream** → Either replay session logs or connect to live sensors.
18. **Design dashboard layout** → Define panels for workload, accuracy probability, and emotion transitions.
19. **Integrate visualizations** → Create gauges, line charts, and emotion flow diagrams.
20. **Enable real-time updates** → Refresh predictions and plots continuously as new data arrives.
21. **Test and refine** → Validate responsiveness, accuracy of model outputs, and usability of the interface.

STEP 5: Evaluation & Testing

- Test with simulated data replay → check smooth updates.
Stress test with faster refresh rates (100ms, 500ms).
Validate interpretability: does workload correlate with task difficulty? Does accuracy probability track real outcomes?

STEP 6: Extensions

- **Multi-user support:** dashboard panel per participant.
- **Alert system:** if workload too high OR accuracy drops below threshold → trigger warning.
- **Replay mode:** visualize stored session logs as if live.
- **Export:** save dashboard outputs as MP4 or session summary CSV.

File & Code Organization

project/

```
|— data/
|   |— EEG.csv
|   |— GSR.csv
|   |— EYE.csv
|   |— TIVA.csv
|   |— session_logs/
|— models/
|   |— workload_model.pkl
|   |— accuracy_model.pkl
|   |— emotion_model.pkl
|— app/
|   |— dashboard.py      # main Streamlit/Dash app
|   |— utils.py          # data stream + feature extraction
|   |— components.py     # plotting helpers
|— README.md
```