Below is a complete Q&A Viva Guide for your ATSA Project – FINAL All answers are in **short, clear point format** Each answer includes **relevant block number** for quick reference Covers **why this method**, **why not others**, **what it means**, **business impact**

Block 1: Data Loading & Preprocessing

Q1: Why use pd.read_csv() and pd.to_datetime()? Why not csv module or datetime.strptime?

- **Block 1**
- pd.read_csv(): Fast, handles headers, types, large files
- pd.to_datetime(format='%d-%m-%Y %H:%M'): Parses DD-MM-YYYY HH:MM correctly in one line
- csv module + strptime: Manual, slow, error-prone for 10k+ rows
- Pandas: Built for data science → faster, safer, standard

Q2: Why set_index('datetime') and sort_index()?

- **Block 1**
- Sets time as index → enables resample(), slicing, plotting by date
- sort_index(): Ensures chronological order (critical for time series)
- Without: Resampling fails, ACF/PACF wrong
- No alternative: Required for all time-based ops

Q3: Why resample('D').sum().fillna(0)? Why not mean or drop?

- **Block 1**
- 'D'.sum(): Converts hourly rides → daily total demand
- fillna(0): Missing day = 0 rides (realistic for low-activity)
- Mean: Adds fake data; Drop: Breaks time continuity
- Business: OLA needs full calendar for planning → 0 is safe

Block 2: Time Series Visualization

Q4: Why show full + weekly subplots? Why not just full plot?

- **Block 2**
- Full plot: Shows stability, no trend
- Weekly subplots: Reveals Mon–Fri high, Sat–Sun low
- Single plot hides day-of-week pattern
- Subplots: Clear, comparable, stakeholder-friendly

Q5: Why limit to first 4 weeks? Why not all 50+ weeks?

- **Block 2**
- 50+ subplots → unreadable, slow
- First 4 weeks: Show typical pattern (repeats)
- Code flexible: n_weeks_to_show can be changed
- Interactive Plotly suggested for all weeks

Q6: Why label days as Mon, Tue... instead of dates?

- **Block 2**
- Makes weekday vs weekend effect **instantly visible**
- Dates require mental calculation
- Business insight: “Weekends drop 30%” → surge pricing logic

---

Block 3–4: Stationarity Tests (ADF & KPSS)

Q7: Why use both ADF and KPSS? Why not just ADF?

- **Block 3 & 4**
- ADF: Null = non-stationary → $p<0.05$ → **reject** → stationary
- KPSS: Null = stationary → $p>0.05$ → **fail to reject** → stationary
- Both agree → **strong evidence**
- One test risks false conclusion

Q8: Result: ADF p≈0.000, KPSS p≈0.10 → What does it mean?

- **Block 4**
- Series is **stationary** (no trend, no unit root)
- No differencing needed → d=0 in SARIMAX
- If non-stationary → use d=1 or D=1

Q9: Why autolag='AIC' and nlags='auto'?

- **Block 3**
- Auto-selects best lag count → avoids bias
- Manual lag: Subjective, can distort p-value
- AIC/KPSS auto: Standard, reliable, data-driven

Block 5: Data Smoothing

Q10: Why 7-day SMA, 3-day WMA, SES(α=0.5)?

- **Block 5**
- **7-day SMA**: Removes weekly noise, shows cycle
- **3-day WMA**: Recent days weighted more → faster response
- **SES α=0.5**: Balances old/new → smooth but adaptive
- Different windows → compare speed vs smoothness

Q11: Why overlay all in one plot?

- **Block 5**
- Direct visual comparison
- Shows: SMA lags, WMA/SES react faster
- Clear insight: Weekly pattern survives all smoothing

Block 6: ACF & PACF

Q12: Why lags=40? Why spikes at 7, 14, 21?

- **Block 6**

- 40 lags = ~6 weeks → covers weekly + monthly
- Spikes at 7,14,21 → **weekly seasonality confirmed**
- Guides: Use m=7 in SARIMAX

Q13: PACF spike at lag 1 → What to do?

- **Block 6**
- Suggests **AR(1)** term → use p=1
- No long memory → no high p

Block 7–8: Train-Test & Baselines

Q14: Why 80/20 time-based split? Why not random split?

- **Block 7**
- Time series: Future predicts future → **no random split**
- Last 20% as test → real-world forecast
- Random split → data leakage → false accuracy

Q15: Why 4 baselines? Why Seasonal-Naive best?

- **Block 7 & 8**
- Mean, Naive, Drift: Simple benchmarks
- **Seasonal-Naive**: Uses last week's same day → captures weekly cycle
- Best among baselines → proves seasonality dominates

Q16: Why MAE, RMSE, MAPE?

- **Block 8**
- **MAE**: Avg error in rides (easy to explain)
- **RMSE**: Punishes big mistakes (ops risk)
- **MAPE**: % error (scale-free)
- All standard → no need for MASE/RMSE-log

Block 9: SARIMAX Modeling

Q17: Why SARIMAX not ARIMA?

- **Block 9**
- ARIMA: No seasonality → flat forecast (you saw!)
- SARIMAX: Adds **weekly season (m=7)** + **exog (weather)**
- Fixes flat line → follows actual ups/downs

Q18: Why order=(1,0,1) and seasonal=(1,1,1,7)?

- **Block 9**
- (1,0,1): From ACF/PACF (lag 1 spike, short memory)
- (1,1,1,7): Weekly pattern, D=1 removes seasonal trend
- enforce_stationarity=False: Allows convergence

Q19: Weather coeffs p>0.6 → Why include?

- **Block 9**
- Included to **test** impact → result: **not significant**
- Insight: Weather doesn't drive demand here
- Future: Replace with **holidays, events**

Q20: Why confidence bands?

- **Block 9**
- Shows **forecast uncertainty**
- Widens over time → don't trust day 30 as much as day 1
- Builds trust with stakeholders

Block 10: Final Insights

Q21: Why say "focus on calendar effects"?

- **Block 10**
- Weather: p>0.05 → no effect

- Weekly pattern: Strong → calendar (holidays, paydays) likely drivers
- Next step: Add binary holiday flag → better accuracy

Q22: Why retrain weekly?

- **Block 10**
- Demand changes: New promos, events, seasons
- Weekly: Captures recent shifts
- Monthly: Too slow; Daily: Overkill

Model Choice: SARIMAX vs Holt-Winters

Q23: MAE: SARIMAX=230.2, Holt-Winters=229.9 → Which is better?

- **Block 9 + Comparison**
- **Holt-Winters wins by 0.3 → negligible**
- Holt-Winters: Simpler, pure seasonality
- SARIMAX: Flexible for future (add holidays)
- **Use Holt-Winters now, SARIMAX later with events**

General Project Questions

Q24: Why not XGBoost or LSTM?

- **All Blocks**
- Only ~450 days → too small for ML
- Needs lag/rolling features → complex
- SARIMAX/Holt-Winters: Interpretable, proven, faster
- ML: Overkill, risk of overfitting

Q25: Business value for OLA?

- **Block 10**
- Accurate daily forecast → right number of drivers

- Weekend dip → reduce idle cars
- Peak hours → surge pricing
- Save fuel, driver pay, improve ETA

Q26: Limitations?

- **Block 10**
- No holiday modeling
- No outliers (strikes, rain)
- Daily only (not hourly)
- Weather not useful here

Q27: Future work?

- **Block 10**
- Add **holiday dummy** in SARIMAX
- Try **Prophet** with holidays
- Build **hourly model**
- Deploy as **API** (Flask/FastAPI)