

```
1 • CREATE DATABASE CANDIDATES
2 ✖ USE CANDIDATES
3 ⚙ CREATE TABLE students (`Candidate Name` VARCHAR(50),`DOB` DATE,`Registration Number` INT,`State` VARCHAR(25),
4 `Maths` INT,`Physics` INT,`English` INT,`GK` INT, `Total Marks` INT, `PFT` CHAR(12), PRIMARY KEY (`Registration Number`));
```



-- Filter out those students who are failed in PFT and failed in any subject That is minimum passing for a subject is 4

- ```
SELECT * FROM students
WHERE PFT = 'Pass'
AND Maths > 3
AND Physics > 3
AND English > 3
AND GK > 3;
```

# 1: IMPORT

```
In [1]: import pandas as pd
import numpy as np
```

```
In [3]: ml = pd.read_csv("cleaned_candidates.csv")
```

```
In [5]: ml
```

|       | Candidate Name    | DOB        | Registration Number | State          | Maths | Physics | English | GK  | Total Marks | PFT  |
|-------|-------------------|------------|---------------------|----------------|-------|---------|---------|-----|-------------|------|
| 0     | Ishaan Verma      | 2005-10-09 | 100086              | Telangana      | 11    | 20      | 17      | 21  | 69          | Pass |
| 1     | Farhan Singh      | 2005-06-29 | 100168              | Uttar Pradesh  | 15    | 21      | 19      | 17  | 72          | Pass |
| 2     | Aarav D'Souza     | 2005-10-17 | 100194              | Tamil Nadu     | 8     | 8       | 9       | 24  | 49          | Pass |
| 3     | Ishaan Chopra     | 2005-09-25 | 100323              | Rajasthan      | 14    | 22      | 4       | 9   | 49          | Pass |
| 4     | Sukhwinder Pillai | 2005-02-11 | 100409              | Madhya Pradesh | 18    | 9       | 14      | 4   | 45          | Pass |
| ...   | ...               | ...        | ...                 | ...            | ...   | ...     | ...     | ... | ...         | ...  |
| 13310 | Aditya Khan       | 2005-02-12 | 999665              | Andhra Pradesh | 19    | 13      | 18      | 15  | 65          | Pass |
| 13311 | Raj Khan          | 2006-04-23 | 999853              | Andhra Pradesh | 16    | 14      | 17      | 10  | 57          | Pass |
| 13312 | Yash Fernandes    | 2005-09-03 | 999898              | Bihar          | 15    | 14      | 5       | 4   | 38          | Pass |
| 13313 | Imran Reddy       | 2005-01-03 | 999907              | Punjab         | 23    | 12      | 22      | 12  | 69          | Pass |
| 13314 | Aditya Pillai     | 2005-11-22 | 999951              | Karnataka      | 4     | 5       | 4       | 11  | 24          | Pass |

13315 rows × 10 columns

## 2: STATE COUNT

```
In [11]: state_counts = ml['State'].value_counts()
print(state_counts)
```

```
State
Bihar                1756
Uttar Pradesh        1599
Rajasthan             1509
Tamil Nadu            876
Andhra Pradesh        738
Madhya Pradesh        720
Punjab                697
Haryana               622
Karnataka              517
Maharashtra           476
Jharkhand              439
West Bengal           413
Odisha                369
Assam                 342
Telangana              337
Chhattisgarh          268
Kerala                263
Gujarat               261
Himachal Pradesh      260
Jammu & Kashmir + Ladakh 254
Delhi                 137
Uttarakhand           96
Meghalaya             81
Manipur               49
Tripura               47
Nagaland              37
Arunachal Pradesh     32
Puducherry            30
Mizoram               25
Andaman & Nicobar      21
Daman & Diu / D&N Haveli 17
Sikkim                14
Goa                   13
Name: count, dtype: int64
```

## 3: STATE WISE CUTOFF

# 1: NAGALAND

```
In [22]: # Cutoff for Nagaland
Nagaland_data = ml[ml['State'] == 'Nagaland']
cutoff = np.percentile(Nagaland_data['Total Marks'], 80)
Nagaland_selected = Nagaland_data[Nagaland_data['Total Marks'] >= cutoff]
print("Nagaland selected candidates:", len(Nagaland_selected))
print("Nagaland marks (approx):", cutoff)
```

Nagaland selected candidates: 9  
Nagaland marks (approx): 41.0

# 2: SIKKIM

```
In [28]: # Cutoff for Sikkim
sikkim_data = ml[ml['State'] == 'Sikkim']
cutoff = np.percentile(sikkim_data['Total Marks'], 40)
sikkim_selected = sikkim_data[sikkim_data['Total Marks'] >= cutoff]
print("Sikkim selected candidates:", len(sikkim_selected))
print("Cutoff marks (approx):", cutoff)
```

Sikkim selected candidates: 8  
Cutoff marks (approx): 39.0

# 3: DAMAN & DIU / D&N HAVELI

```
In [30]: # Cutoff for Daman & Diu / D&N Haveli
Daman_data = ml[ml['State'] == 'Daman & Diu / D&N Haveli']
cutoff = np.percentile(Daman_data['Total Marks'], 20)
Daman_selected = Daman_data[Daman_data['Total Marks'] >= cutoff]
print("Daman selected candidates:", len(sikkim_selected))
print("Daman marks (approx):", cutoff)
```

Daman selected candidates: 8  
Daman marks (approx): 30.0

# 4: ANDAMAN & NICOBAR

```
In [32]: # Cutoff for Andaman & Nicobar
Andaman_data = ml[ml['State'] == 'Andaman & Nicobar']
cutoff = np.percentile(Andaman_data['Total Marks'], 30)
Andaman_selected = Andaman_data[Andaman_data['Total Marks'] >= cutoff]
print("Andaman selected candidates:", len(Andaman_selected))
print("Andaman marks (approx):", cutoff)
```

Andaman selected candidates: 15  
Andaman marks (approx): 41.0

# 5: BIHAR

```
In [34]: # Cutoff for Bihar
Bihar_data = ml[ml['State'] == 'Bihar']
cutoff = np.percentile(Bihar_data['Total Marks'], 80)
Bihar_selected = Bihar_data[Bihar_data['Total Marks'] >= cutoff]
print("Bihar selected candidates:", len(Bihar_selected))
print("Bihar marks (approx):", cutoff)
```

Bihar selected candidates: 360  
Bihar marks (approx): 67.0

# 6: UTTAR PRADESH

```
In [36]: # Cutoff for Uttar Pradesh
UP_data = ml[ml['State'] == 'Uttar Pradesh']
cutoff = np.percentile(UP_data['Total Marks'], 80)
UP_selected = UP_data[UP_data['Total Marks'] >= cutoff]
print("UP selected candidates:", len(UP_selected))
print("UP marks (approx):", cutoff)
```

UP selected candidates: 341  
UP marks (approx): 60.0

# 7: RAJASTHAN

```
In [38]: # Cutoff for Rajasthan
Rajasthan_data = ml[ml['State'] == 'Rajasthan']
cutoff = np.percentile(Rajasthan_data['Total Marks'], 80)
Rajasthan_selected = Rajasthan_data[Rajasthan_data['Total Marks'] >= cutoff]
print("Rajasthan selected candidates:", len(Rajasthan_selected))
print("Rajasthan marks (approx):", cutoff)
```

Rajasthan selected candidates: 317  
Rajasthan marks (approx): 66.0

## 8: TAMIL NADU

```
In [40]: # Cutoff for Tamil Nadu
TN_data = ml[ml['State'] == 'Tamil Nadu']
cutoff = np.percentile(TN_data['Total Marks'], 80)
TN_selected = TN_data[TN_data['Total Marks'] >= cutoff]
print("TN selected candidates:", len(TN_selected))
print("TN marks (approx):", cutoff)
```

TN selected candidates: 187  
TN marks (approx): 67.0

## 9: ANDHRA PRADESH

```
In [42]: # Cutoff for Andhra Pradesh
AP_data = ml[ml['State'] == 'Andhra Pradesh']
cutoff = np.percentile(AP_data['Total Marks'], 80)
AP_selected = AP_data[AP_data['Total Marks'] >= cutoff]
print("AP selected candidates:", len(AP_selected))
print("AP marks (approx):", cutoff)
```

AP selected candidates: 155  
AP marks (approx): 63.0

## 10: MADHYA PRADESH

```
In [44]: # Cutoff for Madhya Pradesh
MP_data = ml[ml['State'] == 'Madhya Pradesh']
cutoff = np.percentile(MP_data['Total Marks'], 84)
MP_selected = MP_data[MP_data['Total Marks'] >= cutoff]
print("MP selected candidates:", len(MP_selected))
print("MP marks (approx):", cutoff)
```

MP selected candidates: 129  
MP marks (approx): 56.0

## 11: PUNJAB

```
In [46]: # Cutoff for Punjab
Punjab_data = ml[ml['State'] == 'Punjab']
cutoff = np.percentile(Punjab_data['Total Marks'], 80)
Punjab_selected = Punjab_data[Punjab_data['Total Marks'] >= cutoff]
print("Punjab selected candidates:", len(Punjab_selected))
print("Punjab marks (approx):", cutoff)
```

Punjab selected candidates: 147  
Punjab marks (approx): 65.0

## 12: HARYANA

```
In [48]: # Cutoff for Haryana
Haryana_data = ml[ml['State'] == 'Haryana']
cutoff = np.percentile(Haryana_data['Total Marks'], 80)
Haryana_selected = Haryana_data[Haryana_data['Total Marks'] >= cutoff]
print("Haryana selected candidates:", len(Haryana_selected))
print("Haryana marks (approx):", cutoff)
```

Haryana selected candidates: 127  
Haryana marks (approx): 66.0

## 13: MAHARASHTRA

```
In [50]: # Cutoff for Maharashtra
```

```
Maharashtra_data = ml[ml['State'] == 'Maharashtra']
cutoff = np.percentile(Maharashtra_data['Total Marks'], 80)
Maharashtra_selected = Maharashtra_data[Maharashtra_data['Total Marks'] >= cutoff]
print("Maharashtra selected candidates:", len(Maharashtra_selected))
print("Maharashtra marks (approx):", cutoff)
```

Maharashtra selected candidates: 101  
Maharashtra marks (approx): 53.0

## 14: KARNATAKA

```
In [52]: # Cutoff for Karnataka
Karnataka_data = ml[ml['State'] == 'Karnataka']
cutoff = np.percentile(Karnataka_data['Total Marks'], 80)
Karnataka_selected = Karnataka_data[Karnataka_data['Total Marks'] >= cutoff]
print("Karnataka selected candidates:", len(Karnataka_selected))
print("Karnataka marks (approx):", cutoff)
```

Karnataka selected candidates: 107  
Karnataka marks (approx): 61.0

## 15: JHARKHAND

```
In [54]: # Cutoff for Jharkhand
Jharkhand_data = ml[ml['State'] == 'Jharkhand']
cutoff = np.percentile(Jharkhand_data['Total Marks'], 82)
Jharkhand_selected = Jharkhand_data[Jharkhand_data['Total Marks'] >= cutoff]
print("Jharkhand selected candidates:", len(Jharkhand_selected))
print("Jharkhand marks (approx):", cutoff)
```

Jharkhand selected candidates: 88  
Jharkhand marks (approx): 62.0

## 16: WEST BENGAL

```
In [56]: # Cutoff for West Bengal
WB_data = ml[ml['State'] == 'West Bengal']
cutoff = np.percentile(WB_data['Total Marks'], 82)
WB_selected = WB_data[WB_data['Total Marks'] >= cutoff]
print("WB selected candidates:", len(WB_selected))
print("WB marks (approx):", cutoff)
```

WB selected candidates: 82  
WB marks (approx): 52.0

## 17: ODISHA

```
In [58]: # Cutoff for Odisha
Odisha_data = ml[ml['State'] == 'Odisha']
cutoff = np.percentile(Odisha_data['Total Marks'], 80)
Odisha_selected = Odisha_data[Odisha_data['Total Marks'] >= cutoff]
print("Odisha selected candidates:", len(Odisha_selected))
print("Odisha marks (approx):", cutoff)
```

Odisha selected candidates: 77  
Odisha marks (approx): 66.0

## 18: ASSAM

```
In [60]: # Cutoff for Assam
Assam_data = ml[ml['State'] == 'Assam']
cutoff = np.percentile(Assam_data['Total Marks'], 80)
Assam_selected = Assam_data[Assam_data['Total Marks'] >= cutoff]
print("Assam selected candidates:", len(Assam_selected))
print("Assam marks (approx):", cutoff)
```

Assam selected candidates: 73  
Assam marks (approx): 61.0

## 19: TELANGANA

```
In [62]: # Cutoff for Telangana
Telangana_data = ml[ml['State'] == 'Telangana']
cutoff = np.percentile(Telangana_data['Total Marks'], 80)
```

```
Telangana_selected = Telangana_data[Telangana_data['Total Marks'] >= cutoff]
print("Telangana selected candidates:", len(Telangana_selected))
print("Telangana marks (approx):", cutoff)
```

Telangana selected candidates: 75  
Telangana marks (approx): 61.0

## 20: CHHATTISGARH

```
In [64]: # Cutoff for Chhattisgarh
Chhattisgarh_data = ml[ml['State'] == 'Chhattisgarh']
cutoff = np.percentile(Chhattisgarh_data['Total Marks'], 80)
Chhattisgarh_selected = Chhattisgarh_data[Chhattisgarh_data['Total Marks'] >= cutoff]
print("Chhattisgarh selected candidates:", len(Chhattisgarh_selected))
print("Chhattisgarh marks (approx):", cutoff)
```

Chhattisgarh selected candidates: 59  
Chhattisgarh marks (approx): 66.0

## 21: KERALA

```
In [66]: # Cutoff for Kerala
Kerala_data = ml[ml['State'] == 'Kerala']
cutoff = np.percentile(Kerala_data['Total Marks'], 80)
Kerala_selected = Kerala_data[Kerala_data['Total Marks'] >= cutoff]
print("Kerala selected candidates:", len(Kerala_selected))
print("Kerala marks (approx):", cutoff)
```

Kerala selected candidates: 59  
Kerala marks (approx): 61.0

## 22: GUJARAT

```
In [68]: # Cutoff for Gujarat
Gujarat_data = ml[ml['State'] == 'Gujarat']
cutoff = np.percentile(Gujarat_data['Total Marks'], 80)
Gujarat_selected = Gujarat_data[Gujarat_data['Total Marks'] >= cutoff]
print("Gujarat selected candidates:", len(Gujarat_selected))
print("Gujarat marks (approx):", cutoff)
```

Gujarat selected candidates: 54  
Gujarat marks (approx): 66.0

## 23: HIMACHAL PRADESH

```
In [70]: # Cutoff for Himachal Pradesh
HP_data = ml[ml['State'] == 'Himachal Pradesh']
cutoff = np.percentile(HP_data['Total Marks'], 79)
HP_selected = HP_data[HP_data['Total Marks'] >= cutoff]
print("HP selected candidates:", len(HP_selected))
print("HP marks (approx):", cutoff)
```

HP selected candidates: 56  
HP marks (approx): 59.0

## 24: JAMMU & KASHMIR + LADAKH

```
In [72]: # Cutoff for Jammu & Kashmir + Ladakh
JK_data = ml[ml['State'] == 'Jammu & Kashmir + Ladakh']
cutoff = np.percentile(JK_data['Total Marks'], 79)
JK_selected = JK_data[JK_data['Total Marks'] >= cutoff]
print("JK selected candidates:", len(JK_selected))
print("JK marks (approx):", cutoff)
```

JK selected candidates: 58  
JK marks (approx): 61.0

## 25: DELHI

```
In [74]: # Cutoff for Delhi
Delhi_data = ml[ml['State'] == 'Delhi']
cutoff = np.percentile(Delhi_data['Total Marks'], 80)
Delhi_selected = Delhi_data[Delhi_data['Total Marks'] >= cutoff]
print("Delhi selected candidates:", len(Delhi_selected))
```

```
print("Delhi marks (approx):", cutoff)
```

Delhi selected candidates: 29

Delhi marks (approx): 67.0

## 26: UTTARAKHAND

```
In [76]: # Cutoff for Uttarakhand
Uttarakhand_data = ml[ml['State'] == 'Uttarakhand']
cutoff = np.percentile(Uttarakhand_data['Total Marks'], 80)
Uttarakhand_selected = Uttarakhand_data[Uttarakhand_data['Total Marks'] >= cutoff]
print("Uttarakhand selected candidates:", len(Uttarakhand_selected))
print("Uttarakhand marks (approx):", cutoff)
```

Uttarakhand selected candidates: 22

Uttarakhand marks (approx): 47.0

## 27: MEGHALAYA

```
In [78]: # Cutoff for Meghalaya
Meghalaya_data = ml[ml['State'] == 'Meghalaya']
cutoff = np.percentile(Meghalaya_data['Total Marks'], 80)
Meghalaya_selected = Meghalaya_data[Meghalaya_data['Total Marks'] >= cutoff]
print("Meghalaya selected candidates:", len(Meghalaya_selected))
print("Meghalaya marks (approx):", cutoff)
```

Meghalaya selected candidates: 17

Meghalaya marks (approx): 55.0

## 28: MANIPUR

```
In [80]: # Cutoff for Manipur
Manipur_data = ml[ml['State'] == 'Manipur']
cutoff = np.percentile(Manipur_data['Total Marks'], 79)
Manipur_selected = Manipur_data[Manipur_data['Total Marks'] >= cutoff]
print("Manipur selected candidates:", len(Manipur_selected))
print("Manipur marks (approx):", cutoff)
```

Manipur selected candidates: 12

Manipur marks (approx): 56.0

## 29: TRIPURA

```
In [82]: # Cutoff for Tripura
Tripura_data = ml[ml['State'] == 'Tripura']
cutoff = np.percentile(Tripura_data['Total Marks'], 90)
Tripura_selected = Tripura_data[Tripura_data['Total Marks'] >= cutoff]
print("Tripura selected candidates:", len(Tripura_selected))
print("Tripura marks (approx):", cutoff)
```

Tripura selected candidates: 6

Tripura marks (approx): 65.0

## 30: ARUNACHAL PRADESH

```
In [84]: # Cutoff for Arunachal Pradesh
Arunachal_data = ml[ml['State'] == 'Arunachal Pradesh']
cutoff = np.percentile(Arunachal_data['Total Marks'], 81)
Arunachal_selected = Arunachal_data[Arunachal_data['Total Marks'] >= cutoff]
print("Arunachal selected candidates:", len(Arunachal_selected))
print("Arunachal marks (approx):", cutoff)
```

Arunachal selected candidates: 7

Arunachal marks (approx): 51.0

## 31: PUDUCHERRY

```
In [86]: # Cutoff for Puducherry
Puducherry_data = ml[ml['State'] == 'Puducherry']
cutoff = np.percentile(Puducherry_data['Total Marks'], 75)
Puducherry_selected = Puducherry_data[Puducherry_data['Total Marks'] >= cutoff]
print("Puducherry selected candidates:", len(Arunachal_selected))
print("Puducherry marks (approx):", cutoff)
```



Puducherry selected candidates: 7  
Puducherry marks (approx): 42.0

## 32: MIZORAM

```
In [88]: # Cutoff for Mizoram
Mizoram_data = ml[ml['State'] == 'Mizoram']
cutoff = np.percentile(Mizoram_data['Total Marks'], 75)
Mizoram_selected = Mizoram_data[Mizoram_data['Total Marks'] >= cutoff]
print("Mizoram selected candidates:", len(Mizoram_selected))
print("Mizoram marks (approx):", cutoff)
```

Mizoram selected candidates: 8  
Mizoram marks (approx): 39.0

## 33: GOA

```
In [90]: # Cutoff for Goa
Goa_data = ml[ml['State'] == 'Goa']
cutoff = np.percentile(Goa_data['Total Marks'], 90)
Goa_selected = Goa_data[Goa_data['Total Marks'] >= cutoff]
print("Goa selected candidates:", len(Goa_selected))
print("Goa marks (approx):", cutoff)
```

Goa selected candidates: 4  
Goa marks (approx): 38.0

## 4: CHECKING DOB FORMAT

```
In [128]: # If DOB is in 'YYYY-MM-DD' format
ml['DOB'] = pd.to_datetime(ml['DOB'], errors='coerce')
```

## 5: SORTING MERIT LIST

```
In [132]: ml_sorted = ml.sort_values(by=['Total Marks', 'DOB', 'Maths', 'Physics', 'English', 'GK'],
ascending=[False, True, False, False, False, False])
```

## 6: STATE WISE SEAT DISTRIBUTION

```
In [135]: state_seat_distribution = {'Bihar': 360, 'Uttar Pradesh': 341, 'Rajasthan': 317, 'Tamil Nadu': 187, 'Andhra Pradesh': 187, 'Madhya Pradesh': 129, 'Haryana': 127, 'Karnataka': 107, 'Maharashtra': 101, 'Jharkhand': 88, 'West Bengal': 82, 'Odisha': 82, 'Telangana': 75, 'Chhattisgarh': 59, 'Kerala': 59, 'Jammu & Kashmir + Ladakh': 58, 'Himachal Pradesh': 56, 'Gujarat': 56, 'Uttarakhand': 22, 'Meghalaya': 17, 'Andaman & Nicobar': 14, 'Manipur': 12, 'Nagaland': 9, 'Sikkim': 8, 'Mizoram': 8, 'Puducherry': 7, 'Tripura': 6, 'Daman & Diu / D&N Haveli': 6, 'Goa': 3}
```

## 7: STATE WISE CUTOFF DICTIONARY

```
In [138]: # Step 4: Cutoff dictionary (your provided one)
state_cutoffs = {'Bihar': 67, 'Uttar Pradesh': 60, 'Rajasthan': 66, 'Tamil Nadu': 67, 'Andhra Pradesh': 63, 'Madhya Pradesh': 61, 'Karnataka': 61, 'Maharashtra': 53, 'Jharkhand': 62, 'West Bengal': 52, 'Odisha': 66, 'Assam': 61, 'Telangana': 61, 'Chhattisgarh': 66, 'Himachal Pradesh': 59, 'Jammu & Kashmir + Ladakh': 61, 'Delhi': 67, 'Uttarakhand': 47, 'Meghalaya': 17, 'Nagaland': 41, 'Arunachal Pradesh': 51, 'Puducherry': 42, 'Mizoram': 39, 'Andaman & Nicobar': 41, 'Daman & Diu / D&N Haveli': 6, 'Goa': 38}
```

## 8: FINAL MERIT LIST

```
In [141]: final_df = pd.DataFrame()

for state, seats in state_seat_distribution.items():
    state_df = ml_sorted[ml_sorted['State'] == state]
    eligible = state_df[state_df['Total Marks'] >= state_cutoffs.get(state, 999)]
    selected = eligible.head(seats)
    final_df = pd.concat([final_df, selected], ignore_index=True)
```

## 9: EXPORT TO CSV

```
In [144]: # Step 6: Export to CSV on E drive
final_df.to_csv(r'E:\indian_navy_merit_list.csv', index=False)
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js