1.Ανάλυση Αναγκών

1.Αναλυτική παρουσίαση των προβλημάτων του πελάτη

🔹 Πρόβλημα 1: Παρακολούθηση Φορμών και Emails

- Περιγραφή: Ο πελάτης λαμβάνει καθημερινά φόρμες από το website με στοιχεία νέων πελατών, καθώς και emails που περιέχουν είτε στοιχεία πελατών είτε τιμολόγια σε μορφή PDF.
- Πρόβλημα: Η παρακολούθηση αυτών των εισερχόμενων δεδομένων γίνεται χειροκίνητα, με κίνδυνο καθυστερήσεων, λαθών και απώλειας πληροφοριών.
- Επιπτώσεις:
  - ❖ Χαμηλή παραγωγικότητα
  - ❖ Αυξημένος χρόνος διαχείρισης
  - ❖ Πιθανότητα μη έγκαιρης ανταπόκρισης σε νέους πελάτες

🔹 Πρόβλημα 2: Εξαγωγή Δεδομένων από Φόρμες και Emails

- Περιγραφή: Τα στοιχεία πελατών (Όνομα, Email, Τηλέφωνο, Εταιρεία, Υπηρεσία ενδιαφέροντος) καταγράφονται χειροκίνητα σε Excel ή Google Sheets.
- Πρόβλημα: Η διαδικασία είναι επαναλαμβανόμενη και επιρρεπής σε ανθρώπινα λάθη.
- Επιπτώσεις:
  - ❖ Καθυστέρηση στην καταχώριση
  - ❖ Ασυνέπεια στα δεδομένα
  - ❖ Δυσκολία στην παρακολούθηση ιστορικού πελατών

🔹 Πρόβλημα 3: Επεξεργασία Τιμολογίων

- Περιγραφή: Τα τιμολόγια αποστέλλονται μέσω email και σε μορφή HTML και περιέχουν οικονομικά στοιχεία (Αριθμός τιμολογίου, Ημερομηνία, Πελάτης, Ποσό, ΦΠΑ).
- Πρόβλημα: Η εξαγωγή των δεδομένων γίνεται χειροκίνητα, με δυσκολία στην αναζήτηση και οργάνωση.
- Επιπτώσεις:
  - ❖ Χρονοβόρα διαδικασία
  - ❖ Δυσκολία στην οικονομική παρακολούθηση
  - ❖ Πιθανότητα λαθών σε φορολογικά στοιχεία

🔹 Πρόβλημα 4: Κεντρική Διαχείριση Δεδομένων

- Περιγραφή: Ο πελάτης επιθυμεί όλα τα δεδομένα (φόρμες, emails, τιμολόγια) να συγκεντρώνονται σε ένα ενιαίο σύστημα (Google Sheets ή Excel).
- Πρόβλημα: Η ενοποίηση δεν είναι αυτοματοποιημένη και απαιτεί συνεχή ανθρώπινη παρέμβαση.
- Επιπλέον απαιτήσεις:
    - Αυτόματη ενημέρωση των αρχείων
    - Δυνατότητα φιλτραρίσματος και αναζήτησης
    - Ειδοποιήσεις (alerts) για νέα δεδομένα
- Επιπτώσεις:
    - ❖ Δυσκολία στην παρακολούθηση και ανάλυση
    - ❖ Περιορισμένη δυνατότητα λήψης αποφάσεων βάσει δεδομένων
    - ❖ Αδυναμία real-time ενημέρωσης

2. Προτεινόμενες τεχνολογίες και εργαλεία

Για την ανάπτυξη του έργου χρησιμοποιήθηκαν οι εξής τεχνολογίες και βιβλιοθήκες:

Backend & Machine Learning:

1. Python – Κύρια γλώσσα προγραμματισμού για το backend και τη διαχείριση δεδομένων.

2. Flask – Μικροπλαίσιο (microframework) για τη δημιουργία web server και API endpoints.

3. Google Sheets & Google Cloud API – Για αποθήκευση, ανάγνωση και συγχρονισμό δεδομένων σε πραγματικό χρόνο.

4. scikit-learn – Για την υλοποίηση μοντέλου μηχανικής μάθησης που προβλέπει την προτεραιότητα των εισερχόμενων μηνυμάτων.

5. Hugging Face Transformers – Για αυτόματη παραγωγή περιλήψεων (summarization) των email με προεκπαιδευμένα μοντέλα NLP.

6. spaCy – Για επεξεργασία φυσικής γλώσσας (NLP), tokenization και καθαρισμό κειμένου.

7. SentenceTransformer – Για δημιουργία embeddings των κειμένων και συσχετίσεις με βάση τη σημασιολογία.

8. PyTorch – Ως framework για την εκτέλεση μοντέλων μηχανικής μάθησης και deep learning.

9. BeautifulSoup (bs4) – Για parsing και εξαγωγή δεδομένων από HTML αρχεία.

10. unicodedata – Για normalization και καθαρισμό χαρακτήρων κειμένου.

11. pandas – Για διαχείριση και ανάλυση δεδομένων σε μορφή DataFrame.

12. NumPy – Για αριθμητικούς υπολογισμούς και πίνακες δεδομένων.

13. Seaborn & matplotlib – Για οπτικοποίηση δεδομένων και στατιστικά γραφήματα.

14. re (Regular Expressions) – Για pattern matching και επεξεργασία κειμένου.

15. os – Για διαχείριση αρχείων και φακέλων στο λειτουργικό σύστημα.

16. pickle & dill – Για αποθήκευση και φόρτωση προκατασκευασμένων μοντέλων μηχανικής μάθησης.

17. gspread & oauth2client – Για επικοινωνία και συγχρονισμό δεδομένων με Google Sheets χρησιμοποιώντας OAuth2 authentication.

Frontend:

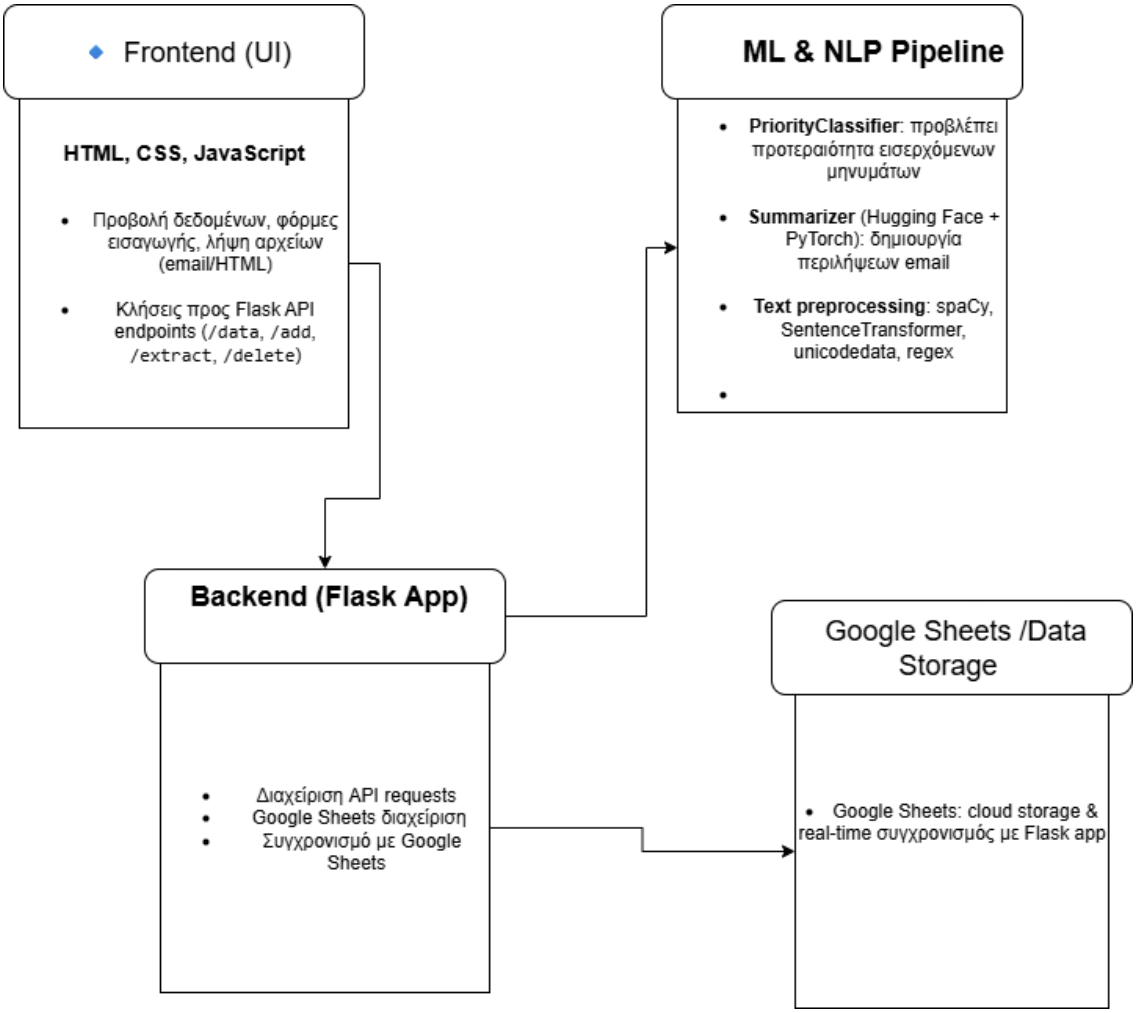
19. HTML – Δομή και markup για τις σελίδες της εφαρμογής.

20. CSS – Στυλ και layout για όμορφο και λειτουργικό περιβάλλον χρήστη.

21. JavaScript – Δυναμική αλληλεπίδραση με τον χρήστη, κλήσεις προς το Flask API και διαχείριση DOM.

Σύντομη Σημείωση: Η συνδυαστική χρήση των παραπάνω τεχνολογιών υποστηρίζει την ανάπτυξη ενός ολοκληρωμένου συστήματος που ενσωματώνει web υπηρεσίες, frontend διεπαφή, NLP, μηχανική μάθηση και διαχείριση δεδομένων σε πραγματικό χρόνο.

3.Αρχιτεκτονική λύσης (διάγραμμα ροής)



Σκοπός Έργου

- Ανάπτυξη web εφαρμογής για αυτοματοποιημένη διαχείριση εισερχόμενων email
- Ταξινόμηση προτεραιότητας, δημιουργία περιλήψεων, εξαγωγή και συγχρονισμός δεδομένων
- Ενσωμάτωση τεχνητής νοημοσύνης για βελτιστοποίηση πληροφορίας και παραγωγικότητας

Αρχιτεκτονική Λύσης

Frontend (UI)

- HTML, CSS, JavaScript
- Προβολή δεδομένων, φόρμες εισαγωγής, λήψη αρχείων (email/HTML)
- Κλήσεις προς Flask API endpoints: `/data`, `/add`, `/delete`, `//list_emails` `/get_file_data/<filename>`,κ.λ.π.

Backend (Flask App)

- Διαχείριση API requests
- CRUD Google Sheets

ML & NLP Pipeline

- PriorityClassifier: προβλέπει προτεραιότητα εισερχόμενων μηνυμάτων
- Summarizer: δημιουργεί περιλήψεις email (Hugging Face + PyTorch)
- Text preprocessing: spaCy, SentenceTransformer, unicodedata, regex

Data Storage

- Google Sheets για cloud storage και real-time συγχρονισμό

Τεχνολογίες και Εργαλεία

- Frontend: HTML, CSS, JavaScript
- Backend: Python, Flask
- NLP/ML: spaCy, Hugging Face Transformers, PyTorch
- Storage: CSV, Excel, Google Sheets API
- Συγχρονισμός: Google Sheets Watcher

4.Οφέλη της Λύσης

- Αυτοματοποιημένη ταξινόμηση και περίληψη email
- Εύκολη εξαγωγή και συγχρονισμός δεδομένων
- Φιλικό περιβάλλον χρήστη για μη τεχνικούς χρήστες
- Με την χρήση Google Sheets μπορούν ταυτόχρονα να διαχειρίζονται τα δεδομένα πολλοι υπάλληλοι

5.Επεκτάσεις και Ασφάλεια

- Προσθήκη authentication layer (OAuth2 ή JWT)
- Logging και monitoring για διαγνωστικούς σκοπούς
- Role-based access για διαχείριση χρηστών και δικαιωμάτων
- Στην υποστήριξη πολλαπλών χρηστών που εκτελούν ταυτόχρονα τη λειτουργία εξαγωγής ("extract") χωρίς καθυστερήσεις, συγκρούσεις ή απώλεια δεδομένων.

2.Τεχνική Πρόταση

1.Λεπτομερής περιγραφή της λύσης

Η προτεινόμενη λύση αποτελεί ένα ολοκληρωμένο σύστημα αυτοματοποιημένης διαχείρισης εισερχόμενων δεδομένων από φόρμες και email, με δυνατότητες ανάλυσης περιεχομένου, εξαγωγής πληροφοριών και συγχρονισμού με cloud-based αποθηκευτικά μέσα. Η υλοποίηση βασίζεται σε web τεχνολογίες, τεχνητή νοημοσύνη και APIs τρίτων υπηρεσιών.

1. Διεπαφή Χρήστη (Frontend)

- Ανάπτυξη UI με HTML, CSS και JavaScript για προβολή και εισαγωγή δεδομένων
- Δυνατότητα λήψης αρχείων (email/HTML) και φόρμες πελατών
- Επικοινωνία με το backend μέσω RESTful API endpoints για αποστολή και ανάκτηση δεδομένων

2. Διακομιστής Εφαρμογής (Backend – Flask)

- Υλοποίηση API endpoints για:
  - `/data`: ανάκτηση δεδομένων
  - `/add`: προσθήκη νέων εγγραφών
  - `/extract`: εξαγωγή περιεχομένου από email ή αρχεία
  - `/export_excel`: δημιουργία αρχείων Excel/CSV
- CRUD λειτουργίες σει Google Sheets
- Διαχείριση locks για αποφυγή ταυτόχρονων εγγραφών και συγκρούσεων

3. Μονάδα Μηχανικής Μάθησης και Επεξεργασίας Κειμένου (ML & NLP Pipeline)

- PriorityClassifier: ταξινόμηση εισερχόμενων email βάσει προτεραιότητας
- Summarizer: δημιουργία περιλήψεων email με χρήση μοντέλων Hugging Face και PyTorch
- Text Preprocessing:
- Καθαρισμός και κανονικοποίηση κειμένου (unicodedata, regex)

4. Αποθήκευση Δεδομένων (Data Storage)

- Google Sheets για cloud συγχρονισμό και real-time ενημέρωση

5. Συγχρονισμός και Ενημέρωση (Synchronization Layer)

- Background thread που παρακολουθεί αλλαγές στο Google Sheets
- Εξασφάλιση συνέπειας και ακεραιότητας των δεδομένων

** Προϋποθέσεις Λειτουργίας

1. Αποθήκευση Αρχείων

Τα αρχεία email και φόρμες αποθηκεύονται είτε σε τοπικό φάκελο είτε σε cloud, ανάλογα με τη διαμόρφωση του συστήματος (δεν αναφέρεται από ποιό μέσο προέρχονται οι φόρμες ή τα email γι αυτό έκανα την συγκεκριμένη προϋπόθεση/απαίτηση)

2. Αλλαγή Τιμών Δεδομένων

Η επεξεργασία των τιμών των ήδη αποθηκευμένων πελατών γίνεται αποκλειστικά μέσω Google-Sheets εφόσον έχουν εισαχθεί σε αυτό, για άμεση και ευέλικτη τροποποίηση των εξαγόμενων πληροφοριών.

2. Χρονοδιάγραμμα Υλοποίησης

Η ολοκλήρωση του έργου προβλέπεται εντός τριών (3) εβδομάδων, βάσει συγκεκριμένων φάσεων με σαφώς ορισμένα παραδοτέα. Το χρονοδιάγραμμα έχει καταρτιστεί ώστε να εξασφαλίζει τεχνική αρτιότητα, επιχειρησιακή ετοιμότητα και δυνατότητα μελλοντικής επεκτασιμότητας.

- **Φάση 1 – Ανάλυση και Σχεδιασμός (1 εργάσιμη ημέρα)**
  Καταγραφή λειτουργικών και τεχνικών απαιτήσεων, σχεδιασμός αρχιτεκτονικής, ορισμός ροών δεδομένων και διασυνδέσεων με εξωτερικά συστήματα.

- **Φάση 2 – Ανάπτυξη Backend (1 εργάσιμη ημέρα)**
  Υλοποίηση RESTful API με Flask, οργάνωση routing, δημιουργία endpoints για εισαγωγή, εξαγωγή και επεξεργασία δεδομένων. Ενσωμάτωση μηχανισμών logging και διαχείρισης σφαλμάτων.

- **Φάση 3 – Ενσωμάτωση AI και Εξωτερικών Υπηρεσιών (1 εργάσιμη ημέρα)**
  Διασύνδεση με Hugging Face για λειτουργίες NLP (συνοψίσεις, κατηγοριοποιήσεις), καθώς και με Google Cloud APIs (OAuth2, Gmail, Sheets) με ασφαλή μηχανισμό αυθεντικοποίησης.

- **Φάση 4 – Ανάπτυξη Διεπαφής Χρήστη (2 εργάσιμες ημέρες)**
  Σχεδιασμός και ανάπτυξη web interface με έμφαση στη χρηστικότητα (usability) και στη βελτιστοποίηση εμπειρίας χρήστη (UX).

- **Φάση 5 – Ενοποίηση και Δοκιμές (2 εργάσιμες ημέρες)**
  Ολοκλήρωση της ενοποίησης, εκτέλεση δοκιμών λειτουργικότητας, απόδοσης και ασφάλειας. Προετοιμασία για παραγωγική λειτουργία.

3. Αναλυτικό Κόστος Υλοποίησης

Η εκτίμηση κόστους έχει βασιστεί σε τεκμηριωμένες χρεώσεις ανά ώρα εργασίας και σε προβλέψιμα λειτουργικά έξοδα, με γνώμονα τη διαφάνεια και την ακρίβεια προϋπολογισμού.

Αρχικό Κόστος Ανάπτυξης

- Ανάπτυξη Flask API & routing: **10 ώρες – €130**
- Ενσωμάτωση Hugging Face μοντέλου: **2 ώρες – €26**
- Ρύθμιση Google Cloud APIs (OAuth, Gmail, Sheets): **2 ώρες – €26**
- Logging, διαχείριση σφαλμάτων & testing: **12 ώρες – €156**
- Σχεδιασμός και ανάπτυξη UI/UX: **12 ώρες – €156**

Συνολικό κόστος αρχικής ανάπτυξης: €494 (38 ώρες)

Μηνιαία Λειτουργικά Έξοδα

- Google Cloud (App Engine, Functions, API calls): **€10**
- Hugging Face API (βάσει τρέχουσας χρήσης): **€0**
- Συντήρηση, debugging & ενημερώσεις: **€100**

Συνολικό μηνιαίο λειτουργικό κόστος: €110

4. Ανάλυση Απόδοσης Επένδυσης (ROI)

Η εφαρμογή στοχεύει στην αυτοματοποιημένη επεξεργασία και σύνοψη εισερχόμενων μηνυμάτων, μειώνοντας δραστικά τον απαιτούμενο χρόνο αξιολόγησης περιεχομένου.

Υποθέσεις Λειτουργίας

- Μέσος όρος εισερχόμενων μηνυμάτων: **100/ημέρα**
- Εξοικονόμηση χρόνου ανά μήνυμα: **2 λεπτά**
- Συνολική εξοικονόμηση: **3,3 ώρες/ημέρα (~66 ώρες/μήνα)**

- Μέση ωριαία αξία εργασίας: **€20/ώρα**

Οικονομική Απόδοση

- Μηνιαίο οικονομικό όφελος: **€1.320**
- Μηνιαίο καθαρό όφελος (μετά λειτουργικών): **€1.210**
- Σημείο απόσβεσης επένδυσης: **~1,5 μήνας**
- Ετήσια καθαρή απόδοση: **>€14.000** (σταθερή χρήση)

Συμπέρασμα: Η επένδυση χαρακτηρίζεται ως **υψηλής απόδοσης**, με ταχεία απόσβεση και σημαντική αύξηση της παραγωγικότητας.

5. Εναλλακτικές Τεχνικές Προσεγγίσεις

Για την ενίσχυση της ευελιξίας και την προσαρμογή σε διαφορετικά επιχειρησιακά περιβάλλοντα, εξετάζονται οι ακόλουθες τεχνικές επιλογές:

Serverless αρχιτεκτονική

Η χρήση υπηρεσιών όπως Google Cloud Functions ή AWS Lambda επιτρέπει την αυτόματη κλιμάκωση και τη μείωση κόστους, με περιορισμούς σε debugging και παραμετροποίηση.

Μετάβαση σε FastAPI

Η αντικατάσταση του Flask με FastAPI προσφέρει καλύτερη απόδοση, native υποστήριξη ασύγχρονων λειτουργιών και ευκολότερη διαχείριση concurrency, με ανάγκη ανασχεδιασμού του backend.

Τοπική φιλοξενία NLP μοντέλων

Η χρήση μοντέλων σε on-premise περιβάλλον εξαλείφει την ανάγκη για εξωτερικά APIs, προσφέροντας πλήρη έλεγχο και μηδενικό λειτουργικό κόστος, με αυξημένες απαιτήσεις σε υποδομή και συντήρηση.

No-code/low-code πλατφόρμες

Η αξιοποίηση εργαλείων όπως Make.com ή Zapier επιτρέπει την ταχεία υλοποίηση βασικών λειτουργιών χωρίς προγραμματισμό, με περιορισμένη δυνατότητα παραμετροποίησης και επεκτασιμότητας.