**✨ Χαρακτηριστικά Εφαρμογής – QuickDrop**

1. **Αιτήματα Παράδοσης από Αποστολείς**

**Οι αποστολείς μπορούν να δημιουργούν αιτήματα παραλαβής και παράδοσης.**

* **Συμπληρώνουν πληροφορίες όπως: περιγραφή πακέτου, σημείο παραλαβής και σημείο παράδοσης.**

1. **Προβολή και Αποδοχή Αιτημάτων από Οδηγούς**

**Οι οδηγοί βλέπουν αιτήματα κοντά τους σε πραγματικό χρόνο.**

* **Έχουν δυνατότητα αποδοχής και ανάθεσης παραγγελίας άμεσα.**

1. **Live Παρακολούθηση Τοποθεσίας Οδηγού**

**Η εφαρμογή καταγράφει και ενημερώνει τη θέση του οδηγού σε πραγματικό χρόνο.**

* **Οι αποστολείς μπορούν να παρακολουθούν την πρόοδο της παράδοσης στον χάρτη.**

1. **Σύστημα Επικοινωνίας (Chat)**

**Υπάρχει δυνατότητα συνομιλίας μεταξύ αποστολέα και οδηγού.**

* **Χρήσιμο για διευκρινίσεις, αλλαγές ή επικοινωνία κατά την παράδοση.**

1. **Υποστήριξη Offline**

**Οι οδηγοί μπορούν να συνεχίσουν να εργάζονται χωρίς σύνδεση στο διαδίκτυο.**

* **Τα δεδομένα αποθηκεύονται τοπικά και συγχρονίζονται αυτόματα μόλις επανέλθει η σύνδεση.**

1. **Push Ειδοποιήσεις**

**Ειδοποιήσεις σε αποστολείς και οδηγούς για:**

* **Νέες παραγγελίες**
* **Ενημέρωση κατάστασης**
* **Νέα μηνύματα στο chat  
  • Υλοποιούνται μέσω Firebase Cloud Messaging (FCM).**

**🛠️ Οδηγός Ανάπτυξης**

**1. Backend: Laravel (API Backend με Sanctum)**

**• Σχεδιασμός RESTful API για εξυπηρέτηση Web SPA (Inertia + React) και React Native App.  
~~• Αυθεντικοποίηση μέσω Laravel Sanctum (session-based για Inertia, token-based για mobile).  
• Δομές Δεδομένων: Χρήστες (senders & drivers), Παραγγελίες, Ειδοποιήσεις, Μηνύματα.~~  
• WebSockets (Laravel Echo + Redis) για real-time ενημερώσεις (π.χ. θέση οδηγού, νέα παραγγελία).  
• Συγχρονισμός δεδομένων για offline χρήση (με timestamps ή versioning).  
• Ενσωμάτωση με Firebase Cloud Messaging (FCM) για push notifications.**

**2. Web Frontend: Inertia.js + React (Single Page Application)**

**• SPA αρχιτεκτονική με React components και Inertia.js για seamless navigation.  
• Αυθεντικοποίηση μέσω Laravel Sanctum (session-based cookies).  
• Responsive UI με Tailwind CSS.  
• Εμφάνιση λίστας παραγγελιών, κατάσταση σε πραγματικό χρόνο (WebSockets).  
• Offline caching με service workers (π.χ. Workbox).  
• Push notifications για νέες παραγγελίες και status ενημερώσεις μέσω FCM.**

**3. Mobile App: React Native**

**• Login/Register μέσω Laravel Sanctum (token-based auth).  
• Dashboard για οδηγούς: λίστα παραγγελιών, δυνατότητα αποδοχής.  
• Live tracking μέσω υπηρεσιών τοποθεσίας (π.χ. Background Location API).  
• Chat μεταξύ αποστολέα και οδηγού.  
• Τοπική αποθήκευση με SQLite ή AsyncStorage.  
• Συγχρονισμός δεδομένων όταν υπάρχει σύνδεση.  
• Native push notifications μέσω React Native Firebase.**

**4. Στρατηγική Offline & Συγχρονισμού**

**• Web: Χρήση IndexedDB ή localStorage για προσωρινή αποθήκευση.  
• Mobile: SQLite ή WatermelonDB για local-first λειτουργία.  
• Ανίχνευση και συγχρονισμός αλλαγών με API versioning.  
• Προτεραιότητα σε “last update wins” ή manual conflict resolution όπου χρειάζεται.**

**5. Push Notifications (FCM)**

**• Χρήση Firebase Cloud Messaging για αποστολή ειδοποιήσεων από το backend.  
• Web: Υλοποίηση push notifications μέσω browser service workers.  
• Mobile: Native FCM integration μέσω React Native Firebase.  
• Ειδοποιήσεις για:**

* **Νέα παραγγελία**
* **Αποδοχή παραγγελίας**
* **Αλλαγή κατάστασης**
* **Νέα μηνύματα chat**