To add features for navigation and nearby vehicle availability to your app, you'll need to incorporate several technologies and algorithms:

**Technologies:**

- **Mapping API:**
  - **Google Maps API:** This is a popular choice for providing maps, directions, and location information.
  - **Apple Maps API:** If targeting iOS specifically, Apple Maps API offers similar functionalities.
  - **Here Maps API:** Another option with extensive mapping capabilities.
- **Location Services:**
  - **GPS:** For accurate location determination.
  - **Network-based location:** As a fallback when GPS is unavailable.
- **Directions API:**
  - **Google Maps Directions API:** Calculates routes, distances, and travel times.
  - **Apple Maps Directions API:** Similar functionality for iOS.
- **Nearby Places API:**
  - **Google Places API:** Finds nearby businesses, places, and points of interest.
  - **Apple Maps Places API:** Similar functionality for iOS.
- **Real-time Traffic API:**
  - **Google Maps Real-time Traffic API:** Provides real-time traffic information for route planning.
- **Ride-hailing API:**
  - **Uber API:** Integrate with Uber's platform for ride-hailing services.
  - **Lyft API:** Another popular ride-hailing platform to integrate with.
  - **Ola API:** For regional ride-hailing services in India.
- **Other Vehicle API:**
  - **Rent a Car API:** For car rental services.
  - **Public Transport API:** For bus, train, and metro schedules and routes.

**Algorithms:**

- **Route calculation:**
  - Dijkstra's algorithm or A* search for finding the shortest path between two points.
  - Consider factors like traffic conditions, distance, and travel time.
- **Nearby places search:**
  - Proximity search algorithms to find places within a specified radius.
  - Use geocoding to convert addresses into latitude and longitude coordinates.
- **Vehicle availability:**
  - Query ride-hailing APIs or other vehicle services to check for available vehicles in a specific area.
  - Use real-time traffic data to estimate travel times and adjust availability accordingly.

**Programming Languages:**

- **Frontend:**

  - JavaScript (with frameworks like React, Angular, or Vue) for building the user interface.
  - Swift or Objective-C for iOS apps.
  - Kotlin or Java for Android apps.
- **Backend:**
  - Node.js (with Express.js or other frameworks) for server-side logic and API interactions.
  - Python (with Django or Flask) is another popular choice.
- **Mobile development:**
  - Swift or Objective-C for iOS.
  - Kotlin or Java for Android.
  - React Native or Flutter for cross-platform development.

**Additional Considerations:**

- **User interface:** Design a user-friendly interface that allows users to easily input their destination, view available routes, and book vehicles.
- **Data storage:** Consider storing user preferences and past search history for personalized recommendations.
- **Error handling:** Implement robust error handling to gracefully handle situations like network failures or API limitations.
- **Testing and optimization:** Thoroughly test your app on various devices and platforms to ensure it works as expected. Continuously optimize performance and user experience.

By combining these technologies and algorithms, you can create a comprehensive navigation app with features for finding directions, nearby vehicles, and booking rides.