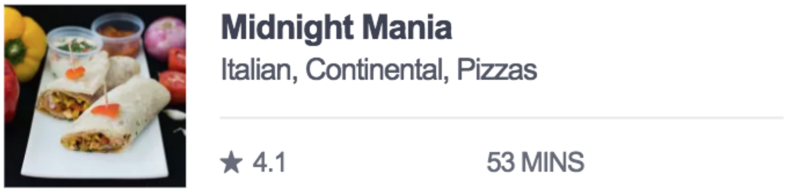
Restaurant Listing

Create a web application that shows a list of restaurants with the following functional & non functional requirements.

Functional Requirements

Restaurant Card

- Each restaurant item should show
  - Image
  - Name
  - Cuisines
  - Rating
  - Delivery Time

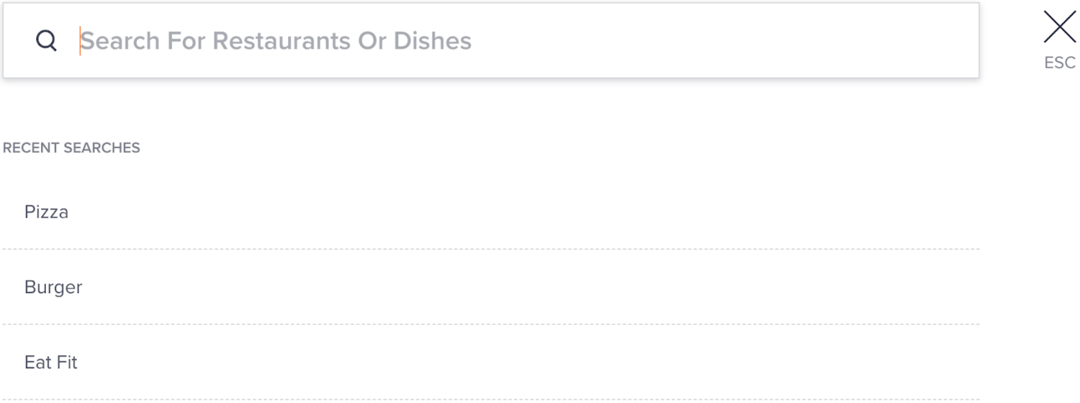


Midnight Mania
Italian, Continental, Pizzas

★ 4.1 53 MINS

Search

- User can search restaurants based on a keyword.
- Search field is equipped with autocompletion, it should be accessible from keyboard.
- Recent searches should be saved.



Search For Restaurants Or Dishes

ESC

RECENT SEARCHES

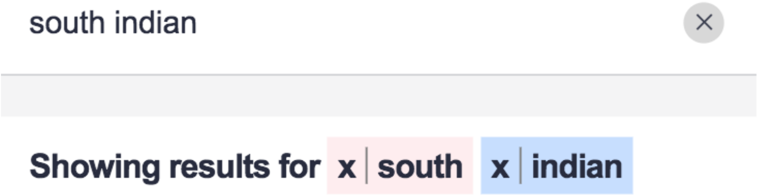
Pizza

Burger

Eat Fit

Sorting & Filtering

- User can sort restaurants based on rating and delivery time.
- Search keywords are split into tags.



south indian

Showing results for x | south x | indian

User Experience

- Data source can be dynamic and refresh at intervals of times. Other features should work reliably with a dynamic source. It's expected for you to mimic the dynamic nature and create a seamless experience to communicate the same to the user.
- Mobile is kept in mind and support of offline mode is there. User should be able to see the list of restaurants and search through it.

Non Functional Requirements

1. Usage of UI frameworks like React, Vue, Angular etc. is permitted.
2. However, prebuilt UI components or libraries aren't allowed. Candidate is expected to implement features & UIs themselves.
3. A server with data API is expected, you may use dummy data (e.g. faker).
4. UI & UX design is upto the candidate. The screenshots are given only to help understand the requirement better.
5. UI should adapt to different resolutions.
6. Code is expected to be scalable, performant and extensible.

For HRs

- UI2 Offsite: All requirements. (Weekend).
- UI2 Onsite: All requirements excluding Sort & Filtering.
- UI1 Offsite: All requirements excluding User Experience. (1 day)
- UI1 Onsite: All requirements excluding Sort & Filtering and User Experience.
- Share the question only after candidate confirms availability and decides to deliver on said date. We don't want candidates to take a weekend/week because they were suddenly busy. If the candidate is busy then let them find time first.

For Reviewers

A checklist is prepared to make our review process as objective as possible.

1. (1 Point) Neat project structure
2. (1 Point) Usage of some CSS convention/standards.
3. (1 Point) Candidate used some ES2015+ features.
4. (1 Point) Code is commented well/type checked/has good variable and component names.
5. (1 Point) Candidate thought of features like SSR, production build, polyfills, tests, SEO etc.
6. (1 Point) Lighthouse performance score is good (> 80?).
7. (1 Point) Thought was given to performance on mobile (throttle/cache/virtualization/lazy loading)

Rating formula: $(\sum \text{Points} / n) \times 5$