

Amazin Trips Frontend Developer Practical Assessment

Instructions for Candidates:

1. You have 4 hours to complete this task from the time you start.
2. Focus on meeting the core requirements before attempting any bonus features.
3. If you're unable to complete all requirements, submit what you have accomplished within the time limit.
4. After completing the application, deploy it to a free hosting platform of your choice (Vercel, Netlify, or similar).
5. Create a GitHub repository for your project and include the deployment link in the README.md file.
6. Email the GitHub repository link to hr@amazintrips.com

Submission:

- Create a GitHub repository for your project.
- Deploy your web application to a free hosting platform (Vercel, Netlify, or similar).
- Include a README.md file with:
 - Brief description of your web app
 - Instructions on how to use the web app
 - Deployment link
 - Any assumptions or additional features you implemented
 - If you had more time, what would you improve or add?

Note:

- While deployment is required, it will not be the primary focus of evaluation. The core functionality and code quality of your application are more important.
- Ensure your deployment is accessible to the public for review.

Time Limit: 4 hours from the time you start

Task:

Create a basic React application for a travel destination wishlist. This app will help users keep track of places they want to visit.

Requirements:

1. Destination List:

- Display a list of travel destinations
- Each destination should show: name, country, and a brief description

2. Add Destination:

- Implement a form to add new destinations
- Fields should include: name, country, and description

3. Remove Destination:

- Allow users to remove a destination from the list

4. Mark as Visited:

- Add functionality to mark a destination as visited
- Visually distinguish visited destinations from non-visited ones

5. Filter Destinations:

- Implement a filter to show:
 - a) All destinations
 - b) Only visited destinations
 - c) Only non-visited destinations

6. Responsive Design:

- Ensure the application is usable on both desktop and mobile devices

7. Data Persistence:

- Use localStorage to persist the destination data

Technical Requirements:

- Use functional components and hooks
- Implement proper state management
- Include basic error handling (e.g., form validation)
- Add basic CSS styling (you can use CSS modules, styled-components, or plain CSS)