

**INTERNSHIP REPORT on**  
**FULL STACK DEVELOPMENT WITH MERN**  
**A REPORT SUBMITTED TO JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,**  
**KAKINADA IN PARTIAL FULFILLMENT OF AWARD OF DEGREE**  
**BACHELOR OF TECHNOLOGY**  
**IN**  
**ELECTRONICS AND COMMUNICATION ENGINEERING**



**SUBMITTED BY**

Mr. G.PRAVEENKUMAR (21HU5A0405)  
Mr. P.SIVA GANGA MANIKANTA (20HU1A0438)  
Mr. V.NANI AVINASH (20HU1A0455)  
Mr. M.SAI AMAR (20HU1A0428)  
Mr. L.CHANDRASEKHAR REDDY (20HU1A0427)

**UNDER ESTEEMED GUIDENCE OF**

**Mr. K. MANOJ PAVAN KUMAR** M. TECH, Asst. professor

**CHEBROLU ENGINEERING COLLEGE**

(Approved by AICTE New Delhi, Affiliated to JNTU-Kakinada)

**ACCREDITED NAAC 'A' GRADE, UGC-AUTONOMOUS**

**Door No:2B, Main Road, Chebrolu, Chebrolu, Guntur**

**522212**



# 1.INTRODUCTION

Full stack development refers to the practice of working on both the front end and back end portions of a web application or software product. A "full stack developer" is someone who is proficient in both the client-side and server-side programming languages and frameworks necessary to build and maintain a complete application.

Here's a breakdown of the components typically involved in full stack development:

1. **Front-end Development:** This involves everything that users see and interact with on a website or application. Front-end developers use languages such as HTML, CSS, and JavaScript to create the layout, design, and functionality of the user interface. They also work with frameworks like React, Angular, or Vue.js to build interactive and responsive web applications.
2. **Back-end Development:** The back end of an application consists of the server, database, and application logic. Back-end developers are responsible for building and maintaining the server-side code that powers the application, manages data storage, and handles user authentication and authorization. Common languages and frameworks used in back-end development include Node.js, Python (with frameworks like Django or Flask), Ruby on Rails, and Java (with Spring framework).
3. **Databases and Data Storage:** Full stack developers often work with databases to store and retrieve data for their applications. This involves understanding database management systems (DBMS) like MySQL, PostgreSQL, MongoDB, or Firebase, and knowing how to write queries to interact with the data.
4. **APIs (Application Programming Interfaces):** APIs allow different software systems to communicate and interact with each other. Full stack developers may need to create APIs to allow their front-end and back-end systems to exchange data seamlessly.
5. **DevOps and Deployment:** Full stack developers are often involved in deploying and maintaining the applications they build. This may involve setting up servers, configuring deployment pipelines, monitoring application performance, and troubleshooting issues that arise in production environments.

# FRONT END DEVELOPMENT

Front-end development focuses on building the user interface and user experience of a website or web application. It involves translating design mockups into code that users can interact with directly in their web browsers. Here's an overview of the key components and technologies involved in front-end development:

1. **HTML (Hypertext Markup Language)**: HTML is the standard markup language used to create the structure of web pages. It defines the layout and content of elements on a web page using tags such as `<div>`, `<p>`, `<h1>`.
2. **CSS (Cascading Style Sheets)**: CSS is used to style the visual presentation of HTML elements on a web page. It allows developers to control aspects like colors, fonts, spacing, and layout. CSS can be applied inline, embedded within HTML, or linked from external style sheets.
3. **JavaScript**: JavaScript is a powerful scripting language that adds interactivity and dynamic behavior to web pages. It is used to manipulate HTML and CSS, handle user input, create animations, fetch data from servers, and much more. Popular JavaScript libraries and frameworks include jQuery, React, Angular, Vue.js, and Ember.js.
4. **Responsive Web Design**: With the increasing variety of devices and screen sizes, it's essential to build websites that adapt and respond to different viewport sizes. Responsive web design techniques, such as using flexible layouts and media queries, ensure that web pages look good and function well on desktops, tablets, and smartphones.
5. **Browser Developer Tools**: Modern web browsers come with built-in developer tools that allow developers to inspect and debug HTML, CSS, and JavaScript code directly in the browser. These tools also provide performance profiling, network monitoring, and other helpful features for front-end development.
6. **Version Control Systems**: Version control systems like Git are essential tools for managing and collaborating on front-end codebases. They allow developers to track changes, revert to previous versions, and work on different branches of a project simultaneously.
7. **Build Tools and Task Runners**: Front-end development often involves tasks such as compiling Sass or Less into CSS, minifying and optimizing JavaScript code, and bundling assets for production. Build tools like Webpack, Gulp, and Grunt automate these tasks and streamline the development workflow.
8. **Web Accessibility**: Web accessibility ensures that web content is usable by everyone, regardless of disability or impairment. Front-end developers should follow best practices for semantic HTML, provide alternative text for images, and ensure keyboard navigation and screen reader compatibility.

Front-end development is a dynamic and rapidly evolving field, with new tools and techniques emerging regularly. Staying up-to-date with the latest trends and technologies is essential for front-end developers to create modern, engaging, and accessible web experiences.

## **BACK END DEVELOPMENT**

Back-end development involves building and maintaining the server-side logic and infrastructure of a web application. While front-end development focuses on the user interface and client-side interactions, back-end development handles the behind-the-scenes operations that enable the application to function. Here's an overview of the key components and technologies involved in back-end development:

1. **Server-Side Programming Languages**: Back-end developers use programming languages to write the logic that runs on the server and processes requests from clients. Common server-side languages include:

- **Node.js**: JavaScript runtime built on Chrome's V8 JavaScript engine, enabling developers to use JavaScript for both front-end and back-end development.
- **Python**: A versatile and easy-to-read language often used for web development, with popular frameworks like Django and Flask.
- **Ruby**: Known for its elegant syntax and developer-friendly ecosystem, with the Ruby on Rails framework being widely used for building web applications.
- **Java**: A robust and widely adopted language, especially for large-scale enterprise applications, often used with frameworks like Spring or Jakarta EE.
- **PHP**: A server-side scripting language commonly used for web development, powering popular platforms like WordPress and Drupal.

2. **Web Servers**: Web servers are software applications that handle incoming HTTP requests from clients (such as web browsers) and serve responses. Common web servers used in back-end development include Apache, Nginx, and Microsoft Internet Information Services (IIS).

3. **Databases**: Back-end developers work with databases to store and manage application data. There are various types of databases, including relational databases like MySQL, PostgreSQL, and Microsoft SQL Server, as well as NoSQL databases like MongoDB, Cassandra, and Firebase Firestore. Back-end developers need to understand database design, querying languages (e.g., SQL), and data modeling techniques.

4. **APIs (Application Programming Interfaces)**: APIs are used to define how different software components or systems communicate with each other. Back-end developers often create APIs to expose functionality and data from the server to the client-side code, allowing them to interact with the application's features programmatically.

5. **Authentication and Authorization**: Back-end developers implement mechanisms for user authentication (verifying user identities) and authorization (determining what actions users are allowed to perform). This often involves techniques like password hashing, session management, JSON Web Tokens (JWT), and OAuth for third-party authentication.

6. **Security**: Back-end developers are responsible for implementing security measures to protect the application against common threats like SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and other vulnerabilities. This includes input validation, data sanitization, encryption, and secure communication protocols like HTTPS.

7. **Middleware and Frameworks**: Middleware and frameworks provide pre-built components and abstractions that streamline back-end development. Examples include Express.js for Node.js, Django for Python, Ruby on Rails for Ruby, Spring Boot for Java, and Laravel for PHP.

8. **Deployment and Hosting**: Back-end developers deploy their applications to servers or cloud platforms to make them accessible over the internet. This involves setting up server environments, configuring databases, managing infrastructure, and ensuring scalability, availability, and performance.

Back-end development is essential for building robust and scalable web applications that can handle complex business logic, manage data efficiently, and provide secure and reliable services to users. Collaboration between front-end and back-end developers is crucial for delivering seamless and cohesive user experiences.

:

Some comprehensive books on antenna design may cover topics related to dual-band microstrip patch antennas. Look for titles by authors such as:

Constantine A. Balanis

Warren L. Stutzman and Gary A. Thiele

Robert S. Elliott

### **ResearchGate and Google Scholar:**

These platforms can be useful for finding research papers, theses, and other scholarly works related to your topic. Researchers often share their publications on these platforms.

### **University Libraries:**

University libraries often provide access to a wide range of academic journals and books. If you're affiliated with a university, check your library's catalog or consult with a librarian.

## MONGO DB

MongoDB is a popular NoSQL (non-relational) database management system that is designed for flexibility, scalability, and performance. Unlike traditional relational databases like MySQL or PostgreSQL, MongoDB stores data in flexible, JSON-like documents, making it well-suited for handling unstructured or semi-structured data and for use cases where data schemas may evolve over time.

Here are some key features and characteristics of MongoDB:

1. **Document-Oriented**: MongoDB stores data in documents, which are JSON-like data structures composed of key-value pairs. These documents can have nested structures and arrays, allowing for more complex data models compared to relational databases.
2. **Schemaless**: MongoDB is schemaless, meaning that documents within a collection can have different structures and fields. This flexibility makes it easier to adapt to changing data requirements and iterate on application development without needing to modify a rigid schema.
3. **High Scalability**: MongoDB is designed to scale out horizontally across multiple servers or clusters, allowing it to handle large volumes of data and high traffic loads. It supports sharding, which involves distributing data across multiple shards (partitions), and replica sets, which provide high availability and data redundancy.
4. **Query Language**: MongoDB provides a powerful query language that supports a wide range of operations for reading, writing, updating, and deleting documents. The query language is based on JavaScript and includes features like aggregation pipelines, indexing, geospatial queries, and full-text search.
5. **Indexing**: MongoDB supports various types of indexes to optimize query performance, including single-field indexes, compound indexes (indexing multiple fields), geospatial indexes, text indexes, and others. Indexes can significantly improve query execution speed by enabling efficient data retrieval.
6. **Geospatial Capabilities**: MongoDB includes built-in support for geospatial queries and indexing, making it well-suited for applications that require location-based features such as mapping, location tracking, and spatial analysis.
7. **Aggregation Framework**: MongoDB's aggregation framework allows developers to perform complex data transformations and analysis operations directly within the database. It provides a flexible and expressive syntax for grouping, filtering, sorting, and aggregating data, similar to SQL's GROUP BY and aggregate functions.



8. **\*\*JSON Data Format\*\***: MongoDB's use of JSON-like documents makes it easy to work with data in modern web development environments, where JavaScript is the primary language. This alignment with JavaScript and other popular programming languages simplifies data manipulation and serialization/deserialization processes.

MongoDB is widely used in a variety of industries and applications, including e-commerce, social networking, content management systems, real-time analytics, Internet of Things (IoT), and more. Its flexibility, scalability, and developer-friendly features make it a popular choice for modern web and mobile applications that require flexible data storage solutions.

## CODES

```
<!DOCTYPE html>
```

```
<html lang="en"
  ><head
    ><title>Spotless Hungry Crocodile</title>
    ><meta
      name="viewport"
      content="width=device-width, initial-scale=1.0"
    /><meta charset="utf-8" /><meta
      property="twitter:card"
      content="summary_large_image"
    /><style data-tag="reset-style-sheet">
      html { line-height: 1.15;}body { margin: 0;}* { box-sizing: border-box; border-width:
0; border-style: solid;}p,li,ul,pre,div,h1,h2,h3,h4,h5,h6,figure,blockquote,figcaption { margin:
0; padding: 0;}button { background-color: transparent;}button,input,optgroup,select,textarea
{ font-family: inherit; font-size: 100%; line-height: 1.15; margin: 0;}button,select { text-
transform: none;}button,[type="button"],[type="reset"],[type="submit"] { -webkit-appearance:
button;}button::-moz-focus-inner,[type="button"]::-moz-focus-inner,[type="reset"]::-moz-focus-
inner,[type="submit"]::-moz-focus-inner { border-style: none; padding: 0;}button:-moz-
focus,[type="button"]:-moz-focus,[type="reset"]:-moz-focus,[type="submit"]:-moz-focus
{ outline: 1px dotted ButtonText;}a { color: inherit; text-decoration: inherit;}input { padding:
2px 4px;}img { display: block;}html { scroll-behavior: smooth }</style>
    ><style data-tag="default-style-sheet">
      html {
        font-family: Inter;
        font-size: 16px;
      }

      body {
        font-weight: 400;
        font-style:normal;
        text-decoration: none;
        text-transform: none;
        letter-spacing: normal;
```

```
line-height: 1.15;  
color: var(--dl-color-gray-black);  
background-color: var(--dl-color-gray-white);
```

```
}</style
```

```
><link
```

```
rel="stylesheet"
```

```
href="https://unpkg.com/animate.css@4.1.1/animate.css"
```

```
/><link
```

```
rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;400;500;600;700;800;900&display=swap"
```

```
data-tag="font"
```

```
/><link
```

```
rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Lato"
```

```
data-tag="font"
```

```
/><link
```

```
rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,100;0,300;0,400;0,700;0,900;1,100;1,300;1,400;1,700;1,900&display=swap"
```

```
data-tag="font"
```

```
/><link
```

```
rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Raleway:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap"
```

```
data-tag="font"
```

```
/><link
```

```
rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;400;500;600;700;800;900&display=swap"
```

```
data-tag="font"
```

```
/><link
```

```
rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Raleway"
```

```
data-tag="font"
```

```
/><link
```

```
rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;400;500;600;700;800;900&display=swap"
```

```
data-tag="font"
```

```
/><link
```

```
rel="stylesheet"
```

```

    href="https://fonts.googleapis.com/css2?family=Raleway:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap"
    data-tag="font"
  /><link
    rel="stylesheet"
    href="https://fonts.googleapis.com/css2?family=Raleway:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap"
    data-tag="font"
  /><link
    rel="stylesheet"
    href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,100;0,300;0,400;0,700;0,900;1,100;1,300;1,400;1,700;1,900&display=swap"
    data-tag="font"
  /><link
    rel="stylesheet"
    href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,100;0,300;0,400;0,700;0,900;1,100;1,300;1,400;1,700;1,900&display=swap"
    data-tag="font"
  /><link
    rel="stylesheet"
    href="https://fonts.googleapis.com/css2?family=Raleway:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap"
    data-tag="font"
  /><link
    rel="stylesheet"
    href="https://unpkg.com/@teleporthq/teleport-custom-scripts/dist/style.css"
  /><style>
  @keyframes fade-in-left {
    0% {
      opacity: 0;
      transform: translateX(-20px);
    }
    100% {
      opacity: 1;
      transform: translateX(0);
    }
  }
</style></head>
<>body
  <>div id="app"></div>
  <>script
    data-section-id="navbar"
    src="https://unpkg.com/@teleporthq/teleport-custom-scripts"

```

```
></script></body>
></html>
```

## CSS CODE

```
.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

## JAVA CODE

```
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css';

import Login from './User/Login';
import Navbar from './User/Unav';
import SelectRide from './User/SelectRide';
import Cabs from './User/Cabs';
import Bookcab from './User/Bookcab';
import Addcar from './Admin/Addcar';
import Mybookings from './User/Mybookings';

import { BrowserRouter, Routes, Route } from 'react-router-dom';
import Register from './User/Register';
import Home from './Components/Home';
import Alogin from './Admin/Alogin';
import Users from './Admin/Users';
import Anav from './Admin/Anav';
import Bookings from './Admin/Bookings';
import UserEdit from './Admin/UserEdit';
import Ahome from './Admin/Ahome';
import Acabs from './Admin/Acabs';
import Acabedit from './Admin/Acabedit';
import Uhome from './User/Uhome';

function App() {
  return (
    <div className="App">
      <BrowserRouter>
        <Routes>

          {/* <Route path="/" element={<Home/>} /> */}
          <Route path="/" element={<Login/>} />
          <Route path="/signup" element={<Register/>} />
          {/* <Route path="/unav" element={<Unav/>} /> */}

          {/* Admin */}
          <Route path="/alogin" element={<Alogin/>} />
          <Route path="/anav" element={<Anav/>} />
          <Route path="/ahome" element={<Ahome/>} />
          <Route path="/users" element={<Users/>} />
          <Route path="/useredit/:id" element={<UserEdit/>} />
          <Route path="/bookings" element={<Bookings/>} />
          <Route path="/acabs" element={<Acabs/>} />
          <Route path="/acabedit/:id" element={<Acabedit/>} />
          <Route path="/addcab" element={<Addcar/>} />

        </Routes>
      </BrowserRouter>
    </div>
  );
}
```

```

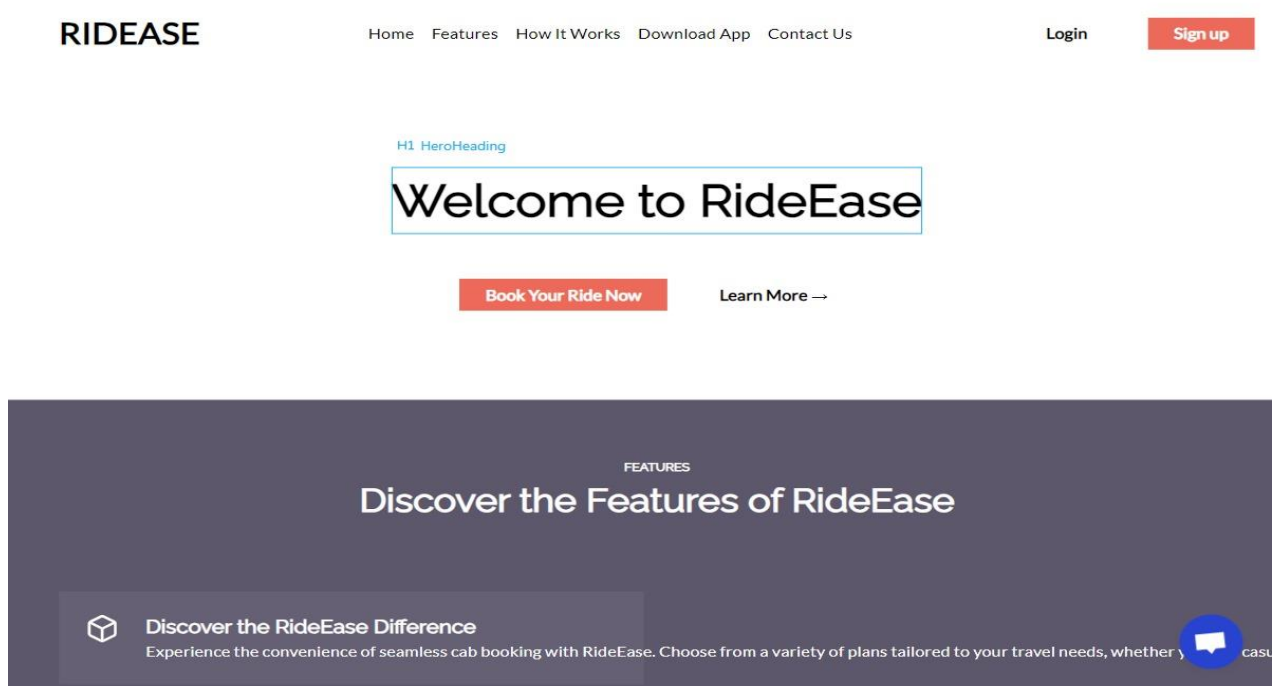
    {/* user      */}
    <Route path='/uhome' element={<Uhome/>}/>
    <Route path='/ucabs' element={<Cabs/>}/>
    <Route path='/bookcab/:id' element={<Bookcab/>}/>
    <Route path='/mybookings' element={<Mybookings/>}/>
  </Routes>
</BrowserRouter>

</div>
);
}

export default App;

```

## RESULT



## Choose Your Plan

[PricingCard](#)

### FREE

Ideal for occasional travelers

\$0

- ✓ Access to standard taxis
- ✓ Limited availability
- ✓ Basic customer support
- ✓ No cancellation fees

[Continue with Free](#)

### BASIC

Perfect for regular commuters

\$5 / month

- ✓ All features of FREE plan
- ✓ Access to standard and premium taxis
- ✓ Increased availability
- ✓ Priority customer support
- ✓ Flexible cancellation policy

[Try the Basic plan](#)

### PRO

Designed for frequent travelers

\$10 / month

- ✓ All features of BASIC plan
- ✓ Access to all vehicle types
- ✓ Guaranteed availability
- ✓ 24/7 premium customer support
- ✓ Exclusive discounts and promotions

[Try the PRO plan](#)

## Simplify Your Travel with RideEase

[Learn More](#)

### FAQ

## Common questions

Here are some of the most common questions that we get.

[Question0](#)

### What types of cars do you sell?

Experience the ultimate convenience and flexibility with RideEase - the innovative cab booking app that puts you in control of your transportation needs. Enjoy seamless booking, real-time tracking, and exclusive rewards for all types of travelers.

## RIDEASE

[Home](#) [Features](#) [How It Works](#) [Download App](#) [Contact Us](#)