

Software Requirements

1. Technology Stack

- **Framework:** The app developed using the Flutter framework.
- **Programming Language:** Developed using the Dart programming language.
- **UI Design:** Utilize Flutter's UI capabilities to create a modern and user-friendly interface.

2. Operating System Requirements

- **Android Version:** Compatible with Android version 6.0 (Marshmallow) or higher.
- **Backward Compatibility:** Ensure the app supports the latest versions of Android while maintaining compatibility with older versions starting from 6.0.

3. Internet Connectivity

- **Network Connection:** An active internet connection (Wi-Fi or mobile data) is required for sharing and viewing resources and placement/internship details.
- **Offline Mode:** Consider implementing an offline mode for certain functionalities such as accessing previously downloaded resources or placement details.

4. User Authentication

- **Authentication:** The app should support user authentication using common methods such as email/password or single sign-on (SSO) methods like Google or Facebook login.
- **Data Protection:** User data must be securely handled and stored according to best practices.

5. Resource Management

- **Uploading Resources:** Allow students to upload notes, PDFs, and other educational materials to the app.
- **File Storage:** Ensure efficient storage and management of uploaded resources.
- **Search & Filter:** Provide robust search and filter options for resources based on metadata such as tags, department, course, and uploader.

6. Grow Together Module

- **Placement/Internship Details:** Enable students to add, edit, and view placement and internship details.
- **Contact Options:** Allow students to contact one another through integrated chat or email functions.
- **Search and Filter:** Provide search and filter capabilities to browse student profiles based on various criteria.

7. Resources Module

- **Uploading:** Students should be able to upload notes, PDFs, and other study materials.
- **Viewing and Rating:** Allow students to view, rate, and comment on resources shared by other students.
- **Sharing:** Enable students to share resources with others within the app.

8. User Interface

- **UI Design:** Implement a clean, intuitive user interface with easy navigation and user-friendly elements.
- **Responsive Design:** Ensure the app is compatible with various screen sizes and orientations.

9. Notifications

- **Push Notifications:** Utilize push notifications to keep students updated on new messages, resource updates, and other app activities.

10. Security and Privacy

- **Data Protection:** Securely store user data, including authentication and uploaded resources.
- **Privacy Controls:** Allow users to manage privacy settings, such as who can view their placement and internship details.

11. Performance

- **Efficiency:** Ensure efficient use of memory and resources for smooth performance.
- **Error Handling:** Implement robust error handling for graceful degradation in case of issues.

Additional Considerations

- **Accessibility:** Ensure the app is accessible to all users, including those with disabilities.
- **Internationalization:** Consider support for multiple languages for a broader user base.
- **User Feedback:** Provide a way for users to offer feedback and suggestions for app improvements.