

COLLEGE CODE : 9623

COLLEGE NAME : Amrita College of Engineering and Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID : 09BB179436342CE839C49558C672F338

ROLL NO : 962323104019

DATE : 28-09-2025

Completed the project named as

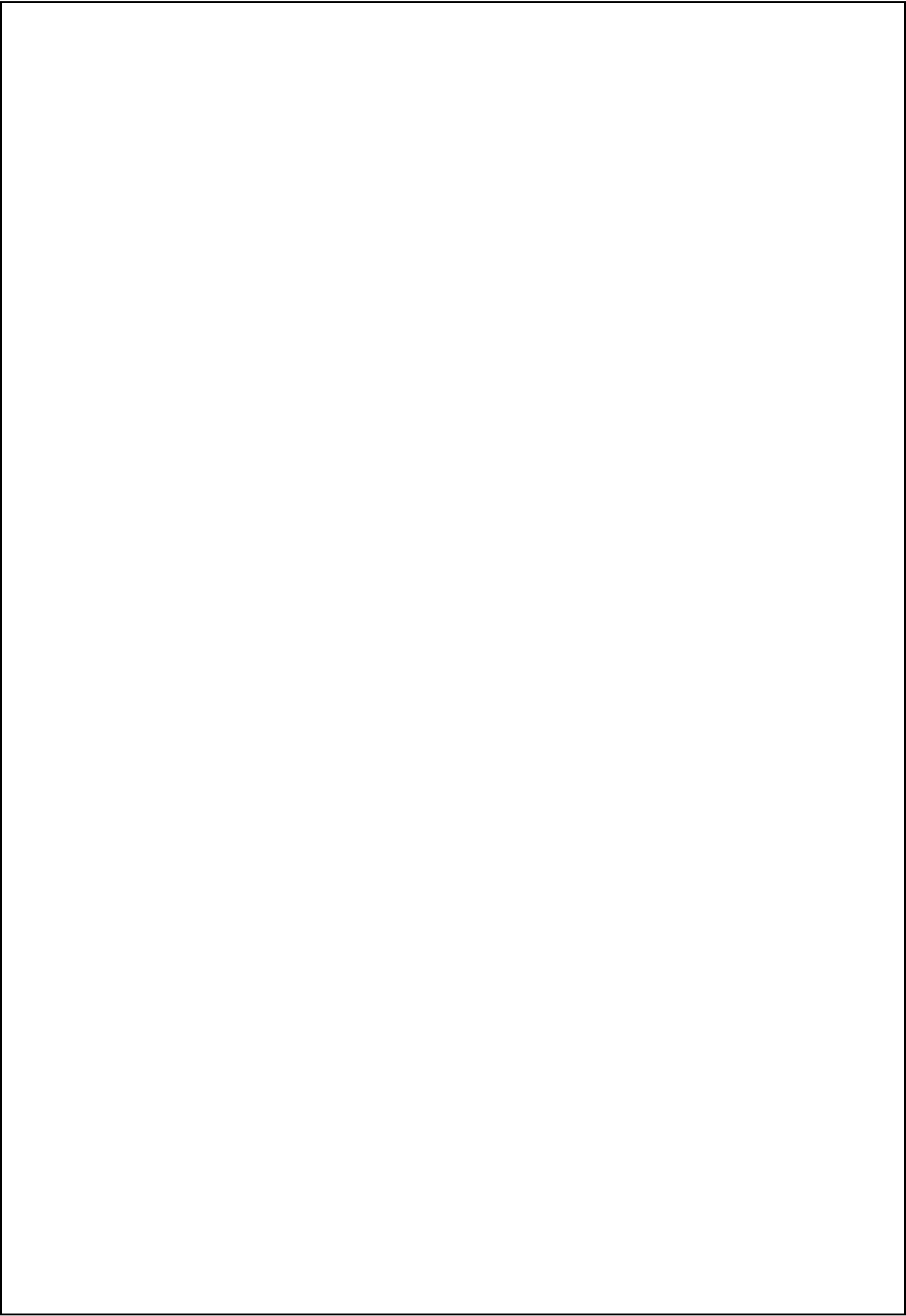
Phase 3

TECHNOLOGY PROJECT NAME : Dynamic Image Slider

SUBMITTED BY,

Name:Arjun.S.S

Mobile no : 8489027136



Phase 3 – MVP Implementation

1. Project Setup

✓ Backend Setup:

- **Tech Stack:** Node.js, Express.js, MongoDB (optional for metadata)
- **Project Structure:**
- backend/
 - | routes/
 - | └─ imageRoutes.js
 - | controllers/
 - | └─ imageController.js
 - | uploads/
 - | app.js
 - └─ package.json
- **Dependencies Used:**
 - express – for API routing
 - multer – for image upload handling
 - cors – for cross-origin support
 - mongoose – if MongoDB is used
 - dotenv – for environment variables

✓ Frontend Setup:

- **Tech Stack:** React.js, Axios
- **Project Structure:**
- frontend/
 - | components/
 - | └─ ImageSlider.jsx
 - | └─ UploadForm.jsx
 - | App.js
 - | index.js
 - └─ package.json
- **Libraries Used:**
 - axios – to make API requests
 - react-slick or custom CSS for slider
 - react-dropzone or basic HTML file input for uploads

2. Core Features Implementation

Feature 1: Image Upload (Admin Only)

- **Frontend:** UploadForm.jsx with file input and Axios POST request.
- **Backend Endpoint:**
- `POST /api/images`
- **Functionality:**
 - Accepts image file and optional title
 - Stores file in `/uploads` folder or cloud (e.g., Cloudinary)
 - Returns uploaded image metadata (URL, title)

Feature 2: Dynamic Image Slider (Frontend)

- **Component:** ImageSlider.jsx
- **Functionality:**
 - Fetches images from backend
 - Displays them in a loop using `setInterval` or `react-slick`
 - Includes navigation arrows or dots (optional)
- **Backend Endpoint:**
- `GET /api/images`

Feature 3: Image Deletion (Optional for Admin)

- **Backend Endpoint:**
- `DELETE /api/images/:id`
- **Frontend:** Admin interface (not visible to normal users)
- **Functionality:** Deletes selected image from backend and refreshes slider

3. API Endpoint Summary

Method	Endpoint	Description
GET	<code>/api/images</code>	Get all uploaded images
POST	<code>/api/images</code>	Upload a new image

Method	Endpoint	Description
DELETE	/api/images/:id	Delete image by ID






4. Sample API Response

```
[
  {
    "id": "1",
    "url": "http://localhost:5000/uploads/image1.jpg",
    "title": "Sample Image",
    "createdAt": "2025-09-25T10:30:00Z"
  }
]
```

5. Implementation Screenshots / Wireframes *(To be added in final report)*

- Slider displaying 3 images
 - Upload form with file chooser and title input
 - Admin view with delete option (optional)
-

6. Testing & Validation

-  Upload works for JPG, PNG formats
 -  Images are served via API and load in slider
 -  Slider transitions smoothly (autoplay every 3 seconds)
 -  Handles empty states (no images)
 -  Optional admin functions tested via Postman
-

7. Conclusion

The MVP version of the Dynamic Image Slider was successfully implemented using a full-stack approach. It allows real-time updates to slider images via a simple backend API. The frontend displays images dynamically, enhancing user engagement and flexibility for various use cases like galleries, portfolios, and product displays.

Next Steps (Post-MVP):

- Add cloud image storage (e.g., Cloudinary)
 - Authentication for admin operations
 - Drag-and-drop image reordering
 - Caption overlay on images
-